SIMULATION AND FABRICATION OF A FORMABLE SURFACE

FOR THE DIGITAL CLAY HAPTIC DEVICE

A Thesis
Presented to
The Academic Faculty

by

Theodore E. Anderson

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Mechanical Engineering in the
School of George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology
May 2007

SIMULATION AND FABRICATION OF A FORMABLE SURFACE

FOR THE DIGITAL CLAY HAPTIC DEVICE

Approved by:

Dr. David W. Rosen, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Wayne J. Book
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Ari Glezer
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: February 12, 2007

To the company of Tyco, I would have never thought about going back to school without the experience I had with them.

ACKNOWLEDGEMENTS

I would like to thank Dr. David Rosen for the opportunity he gave me to work with him. Thank you Dr. Rosen for the support you gave me and for teaching me the proper way to due research.

I would like to thank the other members on my committee, Dr. Wayne Book and Dr. Ari Glezer, for reading my thesis and for the input that they have provided.

Thank you Katrina Trahan for your patience as I completed my research and thesis. Also, thank you for help me with the tedious job of hanging weights from the unit cells and measuring the displacements.

Lastly, I would like to thank my parents, Ted and Joy Anderson, for their unrelenting support and my sister Angel for correcting my poor grammar. I would not have had the courage to quit my job and come back to school without their support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

A formable surface is part of an effort to create a haptic device that allows for a three dimensional human-computer interface called digital clay. As with real clay, digital clay allows a user to physically manipulate the surface into some form or orientation that is sensed and directly represented in a computer model. Furthermore, digital clay will allow a user to change the computer model by manipulating the inputs that are directly represented in the physical model. The digital clay device being researched involves a computer-interfaced array of vertically displacing actuators that is bound by a formable surface. The surface is composed of an array of unit cells that are constructed of compliant spherical joints and translational joints.

As part of this thesis, a series of unit cells were developed and planar surfaces were fabricated utilizing the additive manufacturing process of stereolithography. The process of computing the resultant shape of a manipulated surface was modeled mathematically through energy minimization algorithms that utilized least squares analysis to compute the positions of the unit cells of the surface. Simulation results were computed and analyzed against the movement of a fabricated planar surface.

Once the mathematical models were validated against the manufactured surface, a method for attaching the surface to an array of actuators was recommended.

CHAPTER 1

INTRODUCTION

"Shape is a key element in successful communication, interpretation, and understanding of complex data in virtually every area of engineering, art, science, and medicine. It is a key feature for product design, sculpting, interpreting and understanding complex data and the relationship between geometrical features [1]."

The digital clay project is a NSF funded-multidiscipline effort to create a haptic devise that allows for a three dimensional human-computer interface. The participants of this project are pursuing degrees from the Georgia Institute of Technology in the School of Mechanical Engineering, the School of Electrical Engineering, the College of Computing, and the College of Architecture. Figure 1 shows an illustration of the digital clay human-computer interaction. As with real clay, digital clay allows a user to physically manipulate the surface into some form or orientation. This surface acts as both an input and an output. The orientation of the clay is sensed and the parameters are directly represented in a computer model. Additionally, digital clay will allow a user to change the computer model by manipulating the inputs that are than directly represented in the physical model.

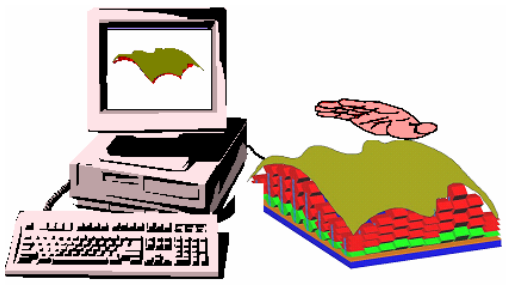


Figure 1 – Illustration of human-computer inaction using a haptic system

The digital clay devise the Georgia Institute of Technology is researching involves the use of hydraulic actuators to alter its shape and to expend an antagonistic force on the hand of a user.

Two different digital clay devices are being pursued. The first clay device is called the bed of nails which is a collection of densely packed actuators that displace vertically and can sense pressure changes. The concept of the bed of nails is shown in Figure 2.

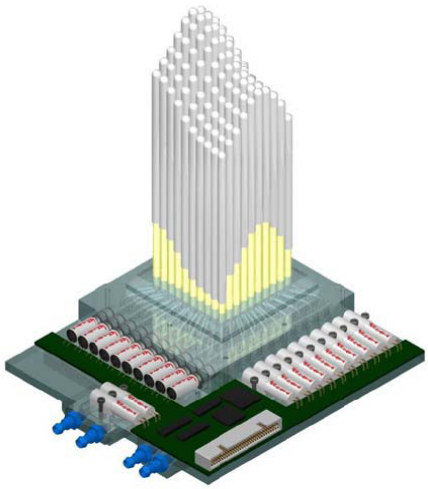


Figure 2 – Illustration of the bed of nails concept

The bed of nails also adjusts the vertical height to replicate a digital model through simulation or received data. In essence, the bed of nails is similar to the "pinhead" 3-D pin sculpture toy depicted in Figure 3, with the digital clay version including an interface via a computer controller [2]. The rods of an ideal bed of nails system are compacted together in a manner that resembles the pixels of a television set. This would enable the bed of nails version of digital clay to produce high resolution objects three-dimensionally. This clay device will not be investigated in thesis.

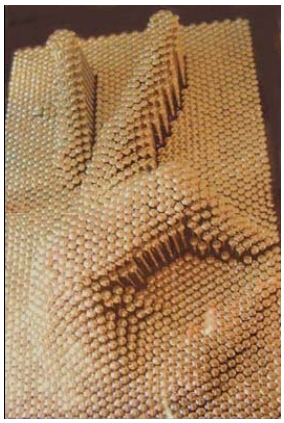


Figure 3 – 3-D pin sculpture toy that is similar to the bed of nails

The second clay device involves a formable crust created by attaching multiple unit cells together to form an array. The terms crust and surface will be used interchangeably in this thesis to describe the formable medium of the digital clay device. Figure 4 depicts the general shape of the unit cells and it shows the common form of the crust that is created from an array of unit cells. The crust is deformed by delivering or removing hydraulic fluid to expandable unit cells [3].

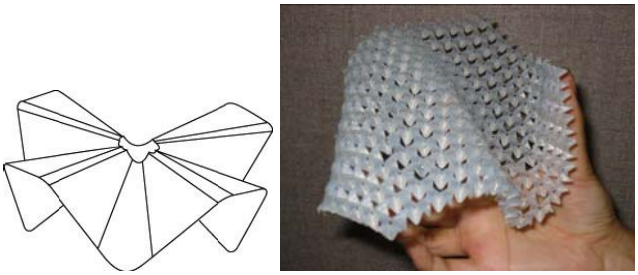


Figure 4 – An example of a unit cell (left) and an array of manufactured unit cells (right)

As fluid is transferred to and from the unit cell, the joint angles will become larger or smaller to distort the shape of the crust. Figure 5 illustrates a unit cell that is expanding due to the delivery of hydraulic fluid. One fabrication method that could be utilized to build these expandable unit cells is micro-electro-mechanical systems (MEMS). Through the use of MEMS technology, the microelectronic integrated circuits are augmented to

allow the microsystems to sense and control the crust. The majority of this thesis will be focused on the formable crust clay device excluding the expandable unit cells and excluding the hydraulic fluid.

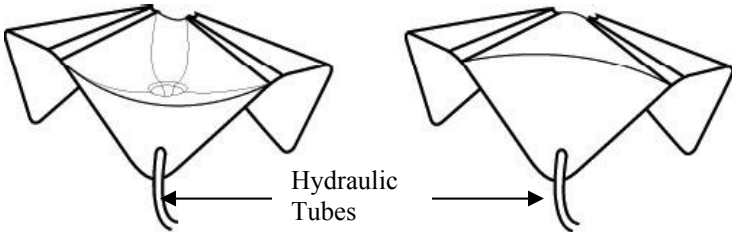


Figure 5 – An example of an expandable unit cell through the use of hydraulics

The technology that is currently available makes it difficult to produce a digital clay device that is primarily actuators or mainly unit cells. The bed of nails device tends to be cumbersome and has poor resolution. MEMS technology is not advanced enough to create thin membranes that could be used to create the unit cells illustrated in Figure 5. A variant of the two different digital clay devices was created to convey the ideas, concepts, and movements of both systems. This variant involves a computer-interfaced bed of nails array that is bound by a formable surface. Once the technology for actuators and MEMS become more advanced, the variant digital clay device can be split into its separate entities. Figure 6 illustrates an example of the formable crust integrated with the bed of nails. An area of interest for this variant involves attaching the surface to the bed of nails without compromising the movement of the surface. This area of interest will be investigated in a later chapter of this thesis.

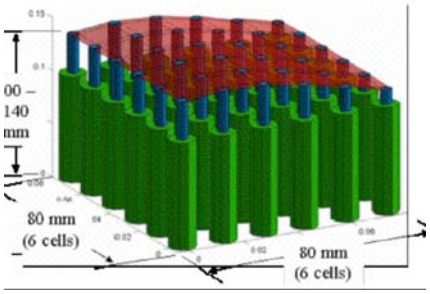


Figure 6 – A display of the variant clay devise that consist of a crust attached to a bed of nails

Device Architecture

The desired digital clay will be a haptic system that will engage the sense of touch of the user by deforming a surface that acts as the haptic interface and the surface bounds a physical volume. The surface is displaced by fluidic-driven actuators that are connected to two common pressurized reservoirs through a dedicated two-way miniature valve [4].

The actuators are comprised of inflatable cells where fluid is delivered by valves and pressure sensors. Figure 7 shows a diagram of the digital clay architecture described within the bottom-dashed box labeled “Device.”

The top dot-dashed box shown in Figure 7, shows the control and interface systems for the digital clay device. The digital clay has two different operational modes in which the user can interact with the clay. The first mode is referred to as the display mode. In this mode, the user manipulates the surface by using a computer. An interface system translates a digital computer representation into commands that will indicate how much fluid flow to deliver to the clay to achieve the desired shape.

The second operational mode is referred to as the shape editing mode. In this mode, the clay is deformed by the user touching the surface. User application of a force onto the surface signals the inflatable cells to either inflate or deflate. A mathematical

model will be used to derive the behavior of the clay. The derived values are then sent to the controller from the actuators.

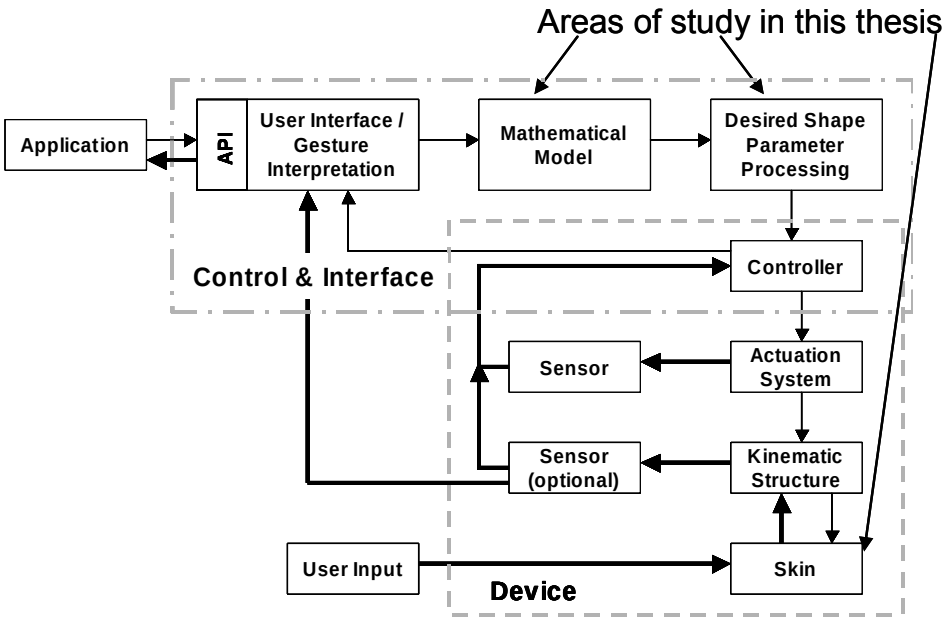


Figure 7 – Digital clay control and device architecture diagram

Applications of the Device

It is important to understand the desire of having a three-dimensional interface like the digital clay project. This project takes a flat two-dimensional (2-D) design and turns it into an actual three-dimensional (3-D) product. As one example, a student at Georgia Tech Lorraine located in France can produce computer images that automatically changes the digital clay surface (the physical model) in their advisor's lab located at Georgia Tech Atlanta. Another example is how digital clay could allow teams working on the same project to have a visual aid that they can touch and manipulate while allowing the other members to experience the changes being made simultaneously. Another great example of an application for digital clay is precisely describing a customized surface geometry. A customer could go to any shoe store and step on a formable surface. The surface would than send data signals of the customer's foot

measurements across the internet to the desired shoe manufacturer for a custom pair of shoes. The digital clay device could be used as an immediately available 3-D prototype. This prototype could be used in a section of machinery and manipulated until the desired functions were met. Computer aided design (CAD) drawings would automatically be produced for this newly designed part because of the computer interface of the digital clay device. Also, digital clay could be used to address the difficulty the visually impaired have with comprehending data generated by computers through haptic perception [5, 6, 7, 8]. The possibility of combining haptic technologies with a Braille system is an additional benefit that could be realized by the visually impaired [9].

Overview of Previous Formable Crust Research

The desire to have a highly deformable crust led to the development of spherical joint mechanisms pictured in Figure 8. These spherical mechanisms address the aspiration for a high degree of freedom to allow the surface to actuate and bend in a multitude of ways [10, 11]. The crust of the clay device consists of arrays of unit cells. These unit cells are comprised of spherical joints and leaf springs. The leaf springs allow the surface to stretch and are the connection point between the unit cells. Figure 8 displays the unit cell and an array that illustrates the connections between the unit cells.

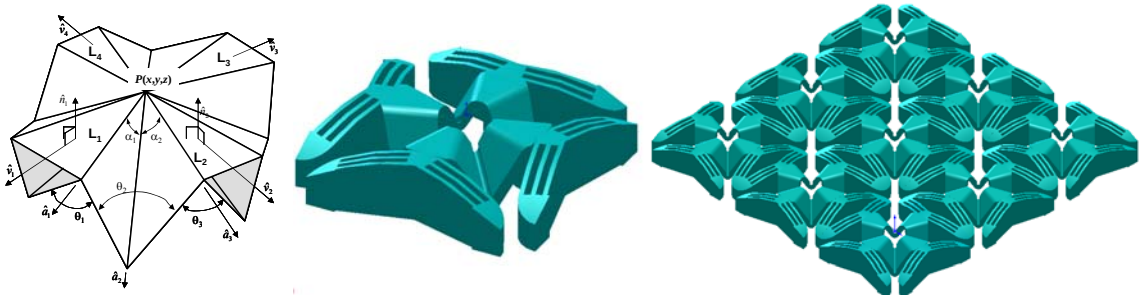


Figure 8 – Model of the spherical joint mechanism (left), model of the unit cell (middle), example of an 3-by-3 array of unit cells (right)

Several algorithms for calculating the deformation of the surface based on given constraints have been created [11, 12]. The algorithms range from calculating regenerated positions of the individual unit cells based on given coordinates and distance constraints to minimizing an energy function that involves the potential energy of the entire crust. Most of the research completed on the algorithms has been theoretical in nature and included a distance constraint. A more comprehensive review of previous work conducted on the formable crust is provided in later chapters of this thesis.

Scope

From the review of previous research, it is observed that the majority of the research has been completed on creating the surface structure or the surface of the digital clay device. Aside from architectural dimensions, structural movement is of significance to a force feedback haptic system. Equally important to the techniques in creating the surface is the computational algorithms describing surface geometry and displacement. No experimental comparison between the algorithms and the manufactured crust has taken place. Also, the current algorithms do not take into account the stretchability of the crust.

The main goal in this thesis was to formulate a computational algorithm that calculates the position of the crust based on material properties and constraints induced on the system. These constraints were the inputs to the algorithm that were used to mimic the positions of the actuators. The objective of the mathematical model was to embody material properties of the surface to simulate the response of the formable crust when constraints were introduced through the means of actuator positions.

Material properties influence the behavior of the formable crust in numerous ways. For instance, the value of the modulus of elasticity, or Young's Modulus, has an influence on the stiffness of the formable crust. A lower value of Young's Modulus will produce a more flexible crust than a higher value. The value of Young's Modulus and

Poisson's ratio for the crust was obtained through a literature review. On the other hand, the structural behavior of the crust was experimentally determined by acquiring the spring constants of the components that form the unit cells. These values were utilized in the mathematical model to obtain an accurate representation of the crust.

Another goal in this thesis was to fabricate a formable crust for the digital clay haptic device. It can be observed from literature that the fabrication of a formable crust has been explored experimentally. No attempt has been made to compare mathematical models to the manufactured crust. An experimental procedure was completed to compare the reaction of a fabricated surface that has been actuated to the estimations produced by the mathematical models. The focus in this thesis was on a planar crust that was actuated by applying a force perpendicular to the crust. The induced system constraints on the planar crust, in the XY plane, with a force being applied in the z-direction allowed movement only in the z-direction.

The last objective of this thesis was to expand on the development of the formable crust by devising a way to integrate the surface with the bed of nails. Two different areas were investigated. The first area involved fabricating the unit cells with additional features that allowed the crust to be attached to the bed of nails. The second area that was investigated involved the use of external fasteners to connect the formable crust to the bed of nails.

Approach to Completing Goals

The main goal in this thesis was to formulate a computational algorithm that calculates the position of the crust based on material properties and constraints induced on the system. Before this goal was achieved, there were a number of tasks that needed to be performed. The tasks are as follows:

- Develop an algorithm to predict the deformation of the formable crust by utilizing previous research and creating mathematical models that apply the spring

representations to the unit cells. Required user inputs should be kept to a minimum, which can cause an under constrained system. The developed algorithm will utilize a numerical iterative process that is capable of determining unit cell positions for an under constrained system.

- Expand on the existing design of the unit cells by creating a unit cell that allows for a more adequate rotation of the spherical joints.
- Manufacture a formable crust using rapid prototyping technology. The desired crust needs to be scalable, formable, and durable.
- Incorporate the material properties of the manufacturing materials into the mathematical models by experimentally determining the joint stiffness for the components of the unit cells.
- Experimentally verify the mathematical models by comparing them to fabricated arrays of the unit cells.

Thesis Organization

The content of this thesis is organized in four main categories. The four main categories are as follows:

- *Background Information* – Chapters 2, 3, and 4 give a summary of the different types of haptic interfaces, the manufacturing process used in this thesis, and an overview of previous research completed on the digital clay project. This overview includes information about the development of the unit cells used to compose the surface of the digital clay device, previous mathematical simulation models, and the idea behind the bed of nails.
- *Simulation models* – Chapters 5, 6, 7, and 8 all relate to the development and testing of additional mathematical models. Chapter 5 introduces the theory behind the additional mathematical models. Chapter 6 is a discussion on the algorithms utilized to solve the mathematical models and some preliminary

testing of the mathematical models. Chapter 7 is an investigation of the theoretical and as-manufactured spring constants for the unit cells. Chapter 8 presents a comparison between the deformation predicted by the mathematical models and the actual deformation occurred on the manufactured surface.

- *Fabricating a working model* – Chapter 9 is an evaluation of various designs for the attachment of the surface to the bed of nails actuating device.
- *Conclusion* – Chapter 10 summarizes the contributions of this research and ends with suggestions for potential future work.

CHAPTER 2

RELATED WORK

A significant amount of research has been accomplished in the area of haptic systems [13]. Many other research groups have developed or are in the process of developing various versions of a haptic interface system [14, 15, 16]. The various haptic system interfaces can be grouped into the following three main categories: exoskeleton, tool handling, and object-oriented. In this chapter, a brief investigation into these three haptic interfaces will be explored.

Exoskeleton Interface

The first haptic interface that will be discussed is the exoskeleton system. Like an exoskeleton on an insect, the idea for this system is to have "muscle attachment" to the outer part of the user. Typically a user wears a glove that can be tracked by the sensors that are attached to the glove [17]. A virtual object can be edited by interpretations of the hand motions by the user. The CyberGrasp devise, depicted in Figure 9, is an example of an exoskeleton system that utilizes a resistive force feedback to each finger [18]. These systems can be bulky because they require additional hardware to implement the force feedback. The ability of humans to sense haptically can be divided into two sub-modalities: the kinesthetic sense, which is activated through force or motion, and the tactile sense, which is activated through tact or touch [19]. Unlike the digital clay project being investigated in this thesis, the CyberGrasp does not enable stimulation of the skin to pressure information by touch. Instead the CyberGrasp relies solely on force being sensed by the receptors in muscles, while the haptic interaction provided by the digital clay device, "offers an independent sensory channel that the brain can process to further enhance a user's experience in a multimodal environment [20]."

Figure 9 – The CyberGrasp exoskeleton haptic system

Tool Handling Interface

Unlike the exoskeleton systems, the tool handling systems use devices that have a point contact interaction with virtual objects. The configuration of these devices is akin to that of a joystick. A user holds a pen-shaped grip that transmits a force in response to the user contacting a virtual object with the pen. One commercially available tool handling system is called the PHANTOM [21, 22], shown in Figure 10.

Figure 10 – The PHANTOM by SensAble Technologies

Another tool handling device is being investigated by a group of research in Japan. The device is like an ordinary pen except it is augmented with haptic and tactile feedback displays. The user utilizes the pen to "push graphical buttons, drag sliders, write and sketch directly on a screen device, just like on a piece of paper [23]." These devices are limited to a virtual environment and do not provide tactile feedback. The digital clay design does not require the user to grip a pen device to achieve a haptic interface but allows the user to involve his or her entire hand in the interaction.

Object Oriented Interface

The object oriented interface allows a user to easily visualize a represented shape and to physically contact it. These mechanisms utilize an array of pins that can manipulate a surface into a desired shape. The digital clay system fits into this interface category because of its ability to physically represent an object.

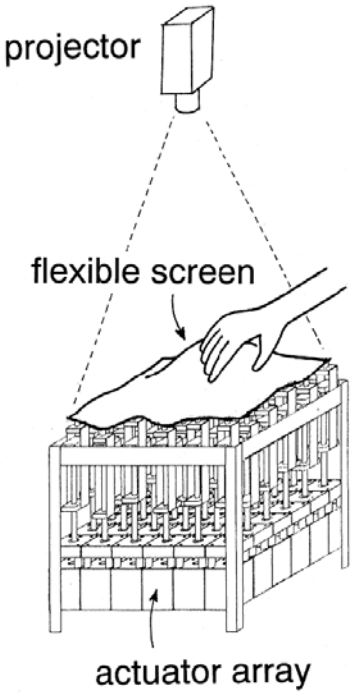


Figure 11 – The FEELEX device

Another example of an object oriented interface is the FEELEX, which is a device that can physically convey a shape to provide both visual and haptic sensations [24]. The word FEELEX is formed by a conjunction of the words feel and flex. This mechanism utilizes an array of linear actuators that deforms its shape. The first concept of the FEELEX, shown in Figure 11, incorporated a rubber screen that was attached to the array of linear actuators. A projector above the apparatus illuminated an image on to the rubber screen as it was deformed by the linear actuators to represent the desired structure.

An object oriented system that utilizes memory alloy wires to provide actuation is being investigated Taylor, Moser, and Creed [25]. The way this system works is by heating a wire by sending a current through it. As the wire heats, it shortens by approximately 5%.

The object oriented interface can also utilize the placement of a surface to create a tactile display that provides the user with the sensation of touching objects. These devices are intended to cause stimulation to the skin by creating patterns with contactors the user touches. Figure 12 illustrates the basic actuator movement for such a system [26].

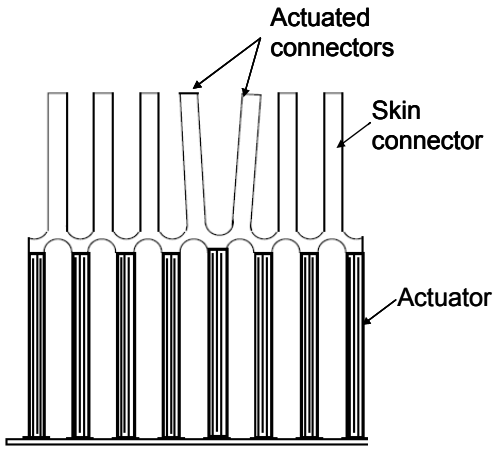


Figure 12 – Illustration of an object oriented interface that provides the user with the sensation of touching an object by swinging the contactors laterally

Miscellaneous Interfaces

Two additional systems that mention the desire to simulate the properties of digital clay are Kizamu and Digital Models from Freehand Sketches. These two systems utilize interactions with a virtual object. In other words, a user can manipulate the shape of a digital representation that is created in a CAD system.

Kizamu

The concept of Kizamu is to meet the demands for high-end digital character design in the entertainment industry to eliminate volume discrepancy in their representations [27]. It is essentially an interactive system for sculpting detailed digital characters. Kizamu does not rely on a physical interaction like the digital clay device being investigated at the Georgia Institute of Technology. Instead, it is a simulation interaction that allows a user to manipulate points of a system model in the computer system.

During the production of Kizamu, three fundamental requirements were identified:

1. The desire for the medium of the system to exhibit the characteristics of real clay.
2. The system needs to run on commonly available hardware that functions at real-time speeds.
3. The system must be compatible with current 3-D representation models and standards.

Adaptively Sampled Distance Fields (ADFs) were introduced to fulfill these requirements and to address the limitations of the volume representations. The definition of a distance field is a scalar field that specifies the minimum distance to the surface of a shape. Through the use of algorithms, the ADFs sample a shape's distance field and stores the

values. The stored distances can then be reconstructed for rendering and sculpting of a digital shape as shown in Figure 13.

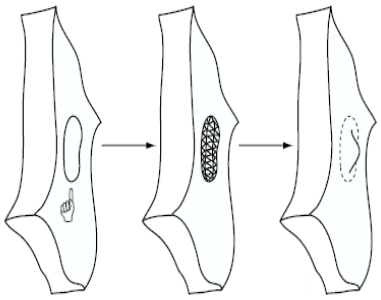


Figure 13 – A region is selected (left), then meshed for control-point editing (middle), ADF regeneration to form desired shape (right)

Digital Models from Freehand Sketches

Unlike the digital clay device being investigate by the team at Georgia Institute of Technology but similar to the Kizamu system, this system is not a physical, haptic interaction between a user and interface. The concept of this system is an interaction between the sketch and the digital model. After the 2-D sketch is purified by the algorithm program, the boundaries and vertices of the drawing are labeled in 3-D orientation and coordinates. Lastly, the drawing is relinquished to a 3-D model in Virtual Reality Modeling Language. Figure 14 shows an example conversion of a 2-D sketch [28].

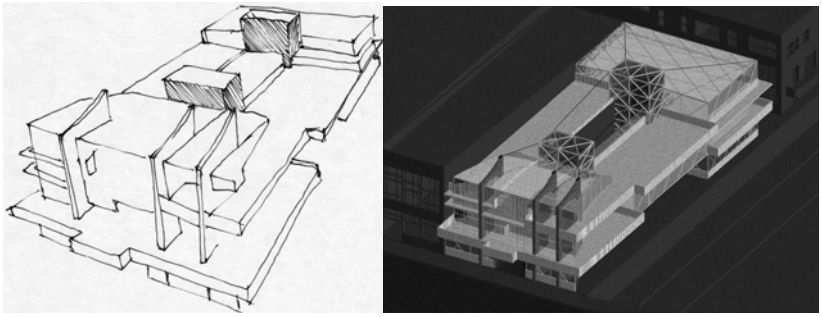


Figure 14 – 2-D hand sketch (left), converted 3-D model (right)

The Digital Clay Haptic Interface System

As mentioned previously, digital clay would fit best in the object oriented interface category because of its ability to morph into a physical object. But unlike the FEELEX haptic device, digital clay will incorporate the capability of producing force feedback for the user without the requirement of extraneous devices like the CyberGrasp or the PHANTOM. In essence, digital clay combines the technologies of the FEELEX and the PHANTOM to produce an advanced haptic system.

CHAPTER 3

ADDITIVE MANUFACTURING

It is important to understand the capabilities of the manufacturing process used to produce the surface of the digital clay device to ensure the design objectives are met. One desire for fabricating the surface is for the parts to be built using scalable manufacturing techniques. The surface of the digital clay device is composed of unit cells with unique formations that would be costly and near impossible to produce from machining. A manufacturing technique that allows scalability and can produce complex shapes is called additive manufacturing. Additive manufacturing is the process of making a part or product by adding layers. Stereolithography, a process of additive manufacturing, was used exclusively for building the surface due to its capabilities of handling complex shapes, the relatively cheap manufacturing cost, and availability of a stereolithography machine. This section summarizes processes of additive manufacturing (AM) that could be used to fabricate the formable crust of the digital clay device with an emphasis on stereolithography to provide an understanding of the manufacturing process utilized in this thesis.

Engineers know first hand how hard it is to visualize a part from a CAD drawing if they have ever inspected one. Improved part visualization can help improve the accuracy of the part. Additive manufacturing is one way of producing a physical part to increase the visualization of an idea which makes this process ideal for fabricating the crust of the digital clay device. The use of additive manufacturing allows designers to discover weaknesses and failure areas early in the design process because a manufactured model is available. The first techniques for additive manufacturing were used to produce models and prototype parts [29]. The words “rapid prototyping” are used to describe additive manufacturing because additive manufacturing was first primarily used to create

prototyped parts. Eventually, rapid prototyping (RP) emerged as a manufacturing process termed rapid manufacturing (RM).

Rapid manufacturing is the use of additive manufacturing technology to directly produce useable products or parts which is the desired for the crust of the digital clay device. In this manufacturing process, RP machines are used as production equipment. With an increase in global competitions, manufactures are challenged to bring new customized products to customers in record time. Rapid manufacturing achieves this demand by shortening the design and production cycles allowing for a quicker release to the market.

Rapid prototyping consists of a combination of a CAD system and a machine that performs the additive manufacturing. AM takes a 3-D CAD representation of a part and reconstructs it into virtual cross sections. These virtual cross sections are generated by slicing the part into thin layers. The virtual cross sections are then sent to the additive manufacturing machine to generate each layer of the part in physical space. The machine manufactures the part in which material is deposited layer by layer with an energy source until the part is completely built. A brief summary of each process that could be used to fabricate the formable crust will be presented.

Stereolithography

As mentioned previously, Stereolithography (SL) was the RP process exclusively used to construct the parts for the research completed in this thesis. A systematic of a SL machine is shown in Figure 15.

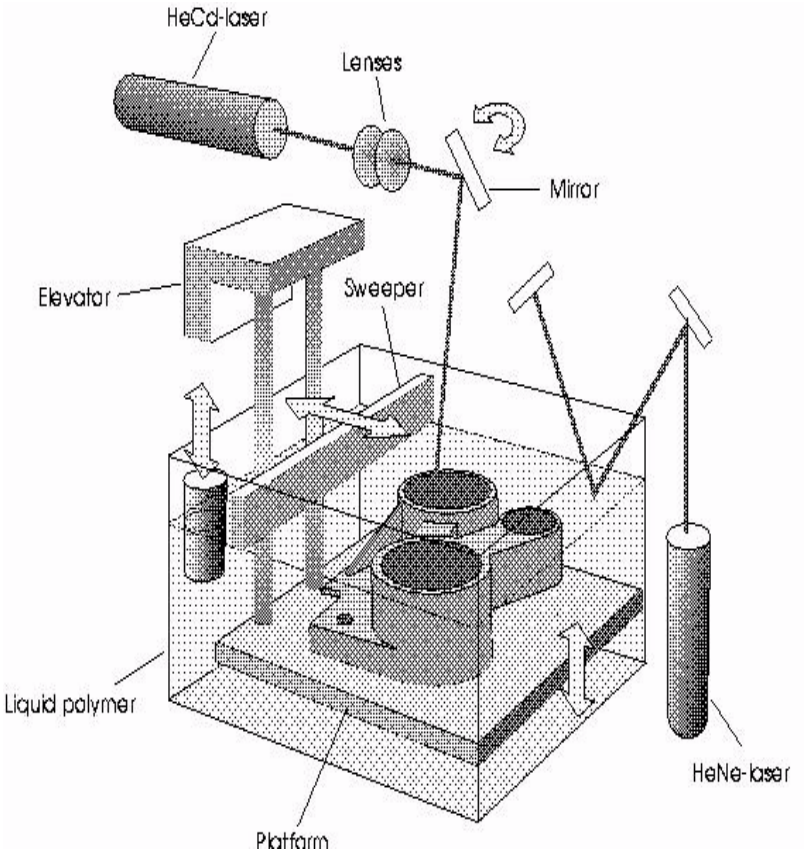


Figure 15 – Schematic of an SL machine

Stereolithography was the first AM machine available commercially that was capable of generating a physical part from a CAD model. It is also one of the most highly used RP technologies available and is produced by a company named 3D Systems. One of the processes involved in SL is to transform a CAD file into an STL or **ST**ereo**L**ithgraphy file [30]. The STL file turns the boundary surfaces of the CAD model into surface triangles.

The next step in the process is to slice the STL file. A series of closely spaced horizontal planes are mapped of the STL file until the whole part is sliced. The closely spaced horizontal planes represent 2D cross-sections of the part.

When the part is completely sliced, it is ready to be fabricated in a SL machine. The SL machines work by tracing the 2D cross-sections, which were created during the slicing process, of the part with a laser in a vat of photopolymer resin. The first step of

the build process in an SL machine involves the creation of a series of support structures to prevent the part from free-floating in the vat of resin. The laser, as shown in Figure 15, is not pointing down into the vat of resin. Instead, the laser beam is redirected by galvanometer-driven mirrors. These mirrors deflect the laser beam into the vat of photopolymer resin. The photopolymer is locally solidified by the ultraviolet beam caused by the laser [31].

After a layer of the photopolymer resin has been cured, the platform holding the part is automatically lowered by the control computer. A sweeper blade slides across the top of the part being fabricated to remove excess resin. The platform and the sweeper blade are shown in Figure 15. Next, the platform is lowered to a position one layer thickness above the previously cured layer. This procedure is repeated until the part is complete.

When the fabrication of the part is complete the excess liquid resin is drained by evaluating the platform out of the vat of photopolymer resin. The support structures are cleaned from the part by an operator once the part is removed from the platform. Now the part is ready for post-processing which will not be discussed.

Selective Laser Sintering

Another RP process that could be utilized to fabricate the surface of the digital clay device is Selective Laser Sintering (SLS), shown in Figure 16. SLS was developed by the University of Texas in Austin and is similar to SL [32]. As with SL, SLS uses a CAD model to physically create a part with a high power laser. Unlike SL, SLS uses a material in a powder bed. SLS machines sinter or melt the powder particles together with the laser. To help control the process and to produce more reasonable build times, the build chamber that contains the powder material is heated uniformly at a temperature just under the melting point of the powder material.

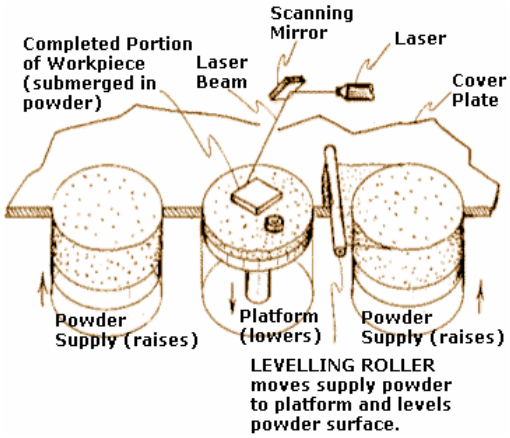


Figure 16 – Schematic of the SLS machine process

One advantage of SLS machines is the wide range of materials that can be used. Some of the powders that have been used include PVC, polycarbonate, nylon, and wax powders. One material in particular is of interest for the fabrication of the crust of the digital clay device. This material, called elastomer, is a flexible material that could allow for the crust to stretch and deform as is desired.

LCVD

Laser Chemical Vapor Deposition (LCVD), shown in Figure 17, is a derivation of the chemical vapor deposition (CVD) process that replaces the global heat source with a localized spot heat source provide by a CO_2 laser. One method of LCVD, called pyrolysis, involves depositing a solid material from a gaseous phase using the CO_2 laser to locally heat a substrate [33]. Gases are delivered into a reaction chamber and passed over or contact the heated substrate. This causes a reaction which forms a solid phase that is deposited onto the substrate. By moving the substrate relative to the position of the laser and gas jet, 3-D structures can be fabricated by layering. Different gases can be delivered into the reaction chamber to produce different deposited material onto the substrate. One major advantage of the LCVD process is that the material is deposited on

the atomic level making the fabricated parts drastically dense. It is possible to get layer thicknesses on the magnitude of a few nanometers.

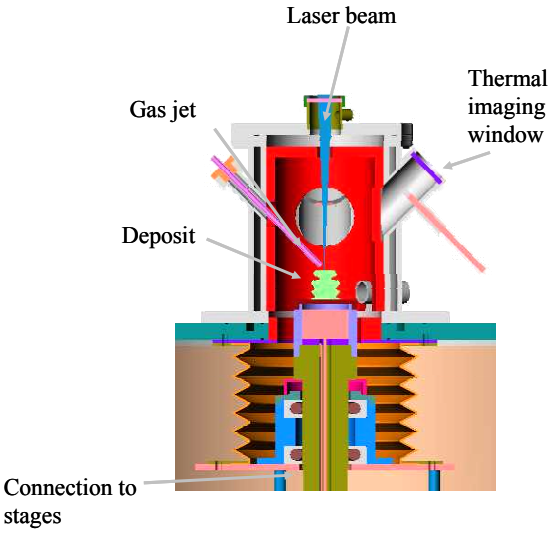


Figure 17 – An example of a LCVD system

The LCVD process is capable of fabricating microstructures. This aspect of LCVD is very desirable for the fabrication of the crust and could possibly be utilized to create a micro-crust for the digital clay device [34].

Micro Stereolithography with Digital Micromirror Display

Micro Stereolithography (SL) with a Digital Micromirror Display (DMD) is a technology that has evolved form the stereolithography technique that was discussed previously. It allows for the manufacture of minuscule parts. Because of this, micro SL is of particular interest in microengineering. Parts are fabricated by superimposing layers obtained by a UV light-induced polymerization of a liquid resin into a solid polymer. A DMD is used to reflect light from a laser onto liquid photopolymer resin in a vat to construct solid microstructures as shown in Figure 18 [35]. Instead of building part layers by drawing its pattern vector-by-vector like conventional SL, micro SL builds an entire layer in one irradiation which allows for rapid fabrication of complex parts. Micro

SL is also capable of producing higher resolution parts than most rapid prototyping process because of its capability to solidify a complete layer. This technique is of particular interest in the microengineering because its 3-D capability allows the production of components no other microfabrication technique can create. Because of this, micro SL is an attractive process capable of creating microstructures that could be utilized to fabricate features on crusts for the digital clay device.

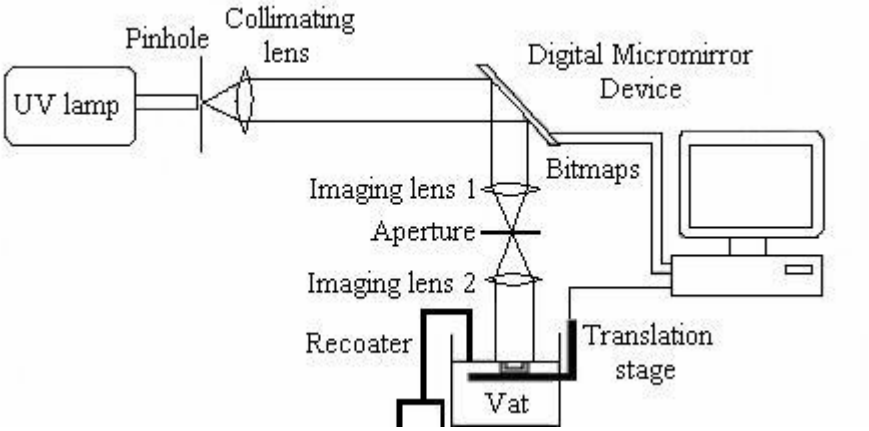


Figure 18 – Schematic of the micro SL system

Crust Fabrication

Most fabrication processes are inadequate to produce the desired flexibility or the complex geometry of the crust. Micromachining is one example of an insufficient manufacturing process. Much of micromachining technology and the majority of the techniques are based on the semiconductor industry. Because of this, micromachining has a limited assortment of veteran materials that can be used. Two widely used materials in this process are silicon and silicon dioxide. Also, the micromachining process tends to be planar in nature and it is not currently conducive to vertical structures because of its premier use in the semiconductor industry. Current research in this area is investigating the use of a flexible epoxy based material called SU-8. More consideration

for the utilization of micromachining in fabricating the crust should be investigated once more extensive knowledge is gained in this area.

Another manufacturing process that is incapable of contriving the unique complex shape of the crust structure is conventional machining. The desire to have miniature sized unit cells and the necessary precision required for the unit cells would prove impractical due to the cumbersome nature of conventional machining. This would cause the fabrication cost and required lead time to be astronomical which is not desirable.

Unlike conventional machining, additive manufacturing (AM) has a high turn around rate and also produces a high level of accuracy. One desire for fabricating the surface is for the parts to be built using a scalable manufacturing technique. Because of the high turn around rate, SL technology can be utilized to fabricate variations of the crust relatively quickly at a low cost. An additional advantage of using stereolithography technology to manufacture the crust is the availability of a SL machine. Although SL is capable of scalable manufacturing, it still has limitations on how small a part can be constructed. Due to the reasons previously mentioned, stereolithography appears to be the most efficient manufacturing method for fabricating the digital clay crust.

CHAPTER 4

UNIT CELL DEVELOPMENT

As described in Chapter 1, the digital clay device is to be a 3-D haptic design device. A critical component of the digital clay device and the main focus of this thesis was the surface to this device. The function of the surface was to act as the medium of contact for the volume of the clay. Before developing the surface, it was import to know what criteria the surface needed to exhibit with respect to the digital clay device.

The first requirement was essential; the surface needed to be malleable in the sense that it can be reshaped and formed. In addition, it was desired for the surface to have a wide range of motion. The surface design needed to be strong enough to withstand the force required to manipulate its shape without breaking. The manufacturability of the complex components that compose the surface was also an important requirement. The surface needed to be manufacturable and the design needed to allow for scaling of the parts.

The structure of the surface was developed with the idea of having a repetitive pattern that incorporated a grid style layout similar to fabrics. The repetitive pattern was used to help manufacture large areas and large quantities that can be scalable. The square-grid crust is a pattern of cells with links attached to neighboring cells' links. The circular points in Figure 19 represent nodal joints and the solid lines represent the links connecting the joints. These links are represented as solid straight lines because they are rigid links. As shown in the figure, the joints can be attached to a maximum of four neighboring joints. The size of the pattern, shown in the figure as a 3-by-3, can be increased to a larger matrix size, for example a 9-by-9 or decreased to a smaller matrix size, for example a 2-by-2.

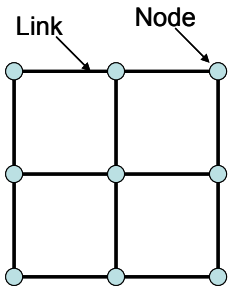


Figure 19 – Schematic of a 3-by-3 square-grid crust pattern

Compliant Spherical Joints

The square-grid crust pattern has a similar grid format as fabric and is easily scalable. The next step was to find a node that has the capability to have a wide range of motion and has the capability of attaching four additional nodes to it. Paul Bosscher completed a considerable amount of research to develop a spherical joint mechanism that works nicely as the node in the square-grid crust pattern [36].

Spherical joints permit three degrees of rotational freedom between two links. For the purpose of creating a surface in the square-grid pattern, it was desired for the joints to be collocated. The spherical joint mechanism developed by Bosscher allows for multiple connections between the main links. Figure 20 show Bosscher's spherical joint mechanism. These main links are connected to one another by intermediate links. These connections between the main links and the intermediate links are produced by revolute joints.

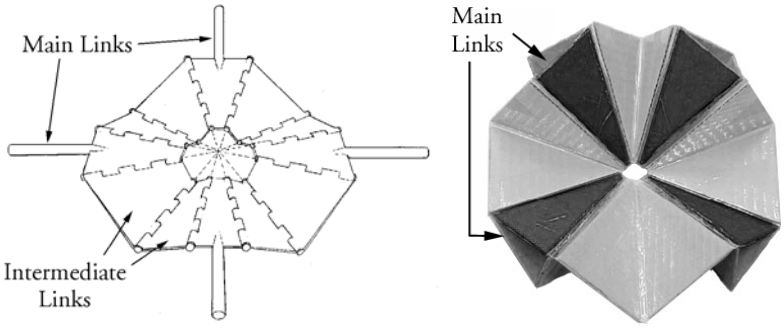


Figure 20 – Diagram of scalable spherical joint mechanism (left) and a manufactured spherical joint mechanism (right)

Austina Nguyen found manufacturing the spherical joints with revolute intermediate joints to be an issue [11]. Nguyen discovered by changing the revolute joints to compliant joints, Bosscher's design can be applied on a smaller scale and manufactured relatively easy. Figure 21 illustrates two plates connected together by a thin flexible joint. This thin flexible joint is an example of a compliant joint.

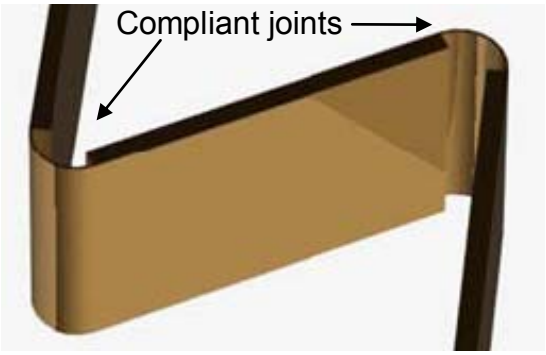


Figure 21 – Example of a compliant joint connecting to plates

The spherical compliant joint, also referred to as the unit cell, illustrated in Figure 20, can be connected at the main links to another unit cell. A 3-by-1 array of unit cells is shown in Figure 22.

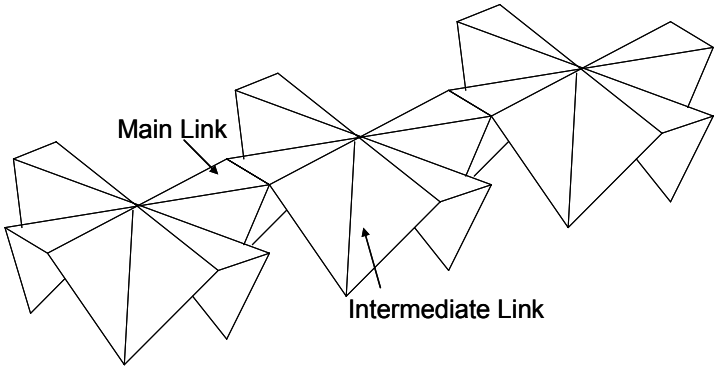
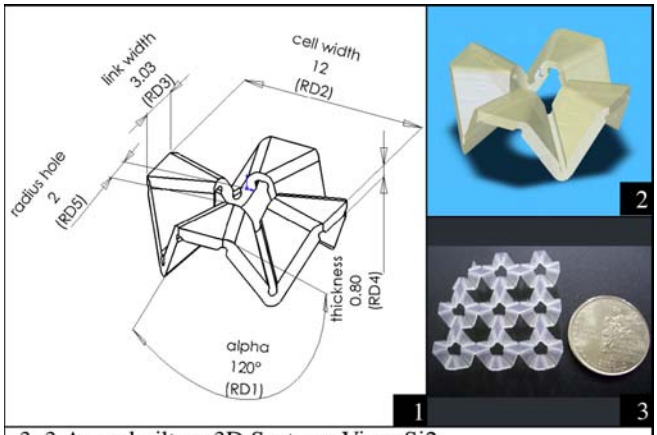


Figure 22 – A 3-by-1 array of spherical joints

A student by the name of Gerrit Becker continued the development of the unit cells and established a parameterized CAD model based on the 45 degree symmetric

characteristic that the unit cells possess. Becker manufactured several different variations of the unit cell design to determine the optimal parameters of the unit cell based on today's technology. The optimal parameters were determined by having a requirement to create a unit cell that is capable of deforming in various directions without breaking and being able to link them together to form a matrix of the cells. Through different experiments, he was able to produce an "optimal" unit cell design [37]. The design that was chosen by Becker is shown in Figure 23.



3x3 Array built on 3D Systems Viper Si2	
Mode: Exact	Room Temp: 70F
Line Width Compensation: Auto	Room Humidity: 61%
Z Correction: Auto	Vat Temp: 25.3C
Resin: SL-7545	Laser Power: 101mW

Figure 23 – Parameters of unit cell design

The designed unit cell shown in Figure 23 has a height to width ratio of about 1:2. This ratio provides long enough intermediate joints to allow bending at the compliant joints while still providing enough stiffness for the unit cell not to bend under its own weight.

Leaf Spring Connections

In addition to finding the parameters of the unit cell, Becker also designed a non-rigid linkage between unit cells. This was an important feature of the digital clay's surface because as the surface is actuated vertically, the rigid surface's projected area

shrinks. The shafts of the actuators are also rigid, which does not allow for any compensation to a stiff crust.

As Becker's research shows, there are three known basic flexible joint components which can provide any degree of freedom. The three flexible joints as shown in Figure 24 are as follows: the planar notch joint, the spherical joint, and the leaf-spring joint [38].

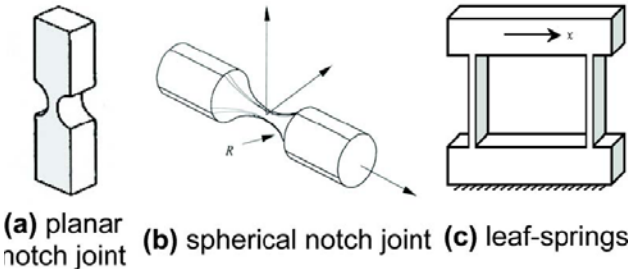
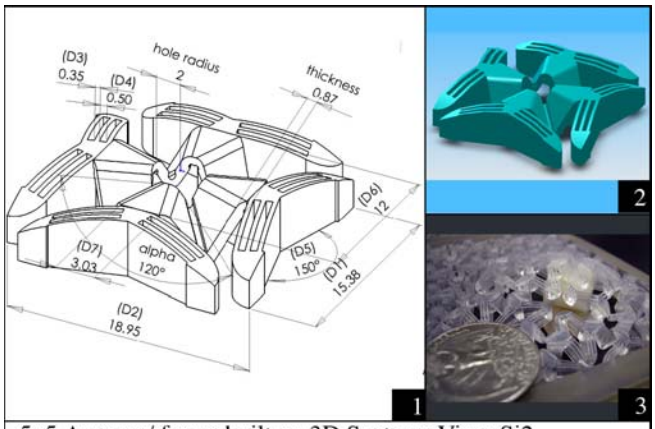


Figure 24 – Examples of the three flexible joints

Through stereolithography, many variations of these flexible joint components were manufactured and tested. Figure 25 shows the design of the unit cell with the flexible joint that was chosen by Becker, a leaf-spring joint. The unit cell design depicted in Figure 25 was the main focus of testing in later chapters. "Leaf springs provide the most generic flexible translational joint, composed of sets of parallel flexible beams [38]." Through his testing, Becker found that this statement holds true and the leaf-spring style of joint allowed for the most horizontal stretch and was still easily manufactured using stereolithography.



5x5 Array w/ frame built on 3D Systems Viper Si2	
Mode: High Res	Room Temp: 70F
Line Width Compensation: Auto	Room Humidity: 58%
Z Correction: Auto	Vat Temp: 25.3C
Resin: SL-7545	Laser Power: 63mW

Figure 25 – Spherical compliant joint unit cell with flexible joints

In summary of the development of the unit cell, the formable surface of the digital clay is comprised of arrays of unit cells. The unit cells were designed to have a high degree of freedom to allow the surface to bend and move in many different ways. A compliant spherical joint made it possible to achieve the desired degree of freedom. To enable the surface to stretch in the horizontal direction, leaf spring connectors were added to the compliant spherical joint. Figure 26 shows a model of the unit cell with leaf springs and an example of an array of unit cells.

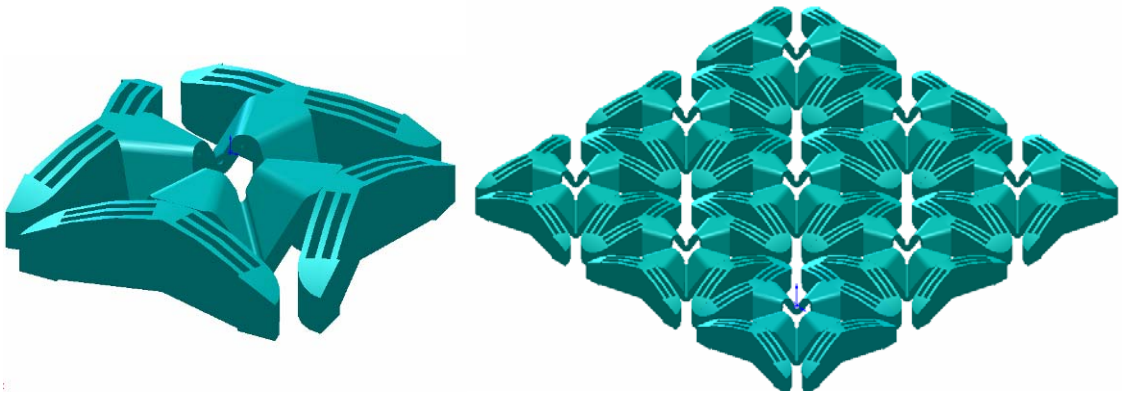
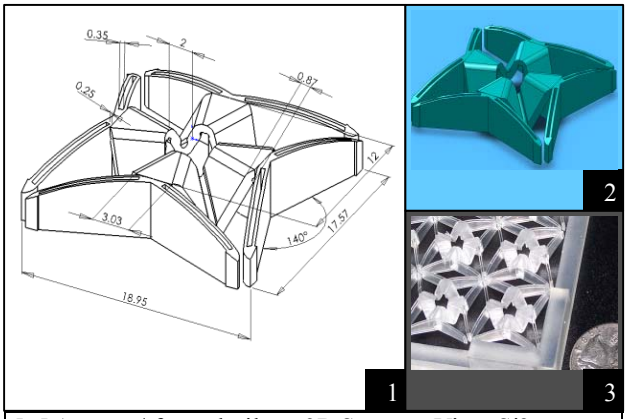


Figure 26 – Model of the unit cell (left), example of an 3-by-3 array of unit cells (right)

Additional Unit Cell Designs

A critical observation was brought to light upon examination of a fabricated 5-by-5 array of the unit cell pictured in Figure 26. It was observed that the leaf spring connectors were limiting the bending capability of the compliant spherical joints. To help eliminate the bending stiffness induced by the leaf spring connectors, four additional unit cell designs were created.

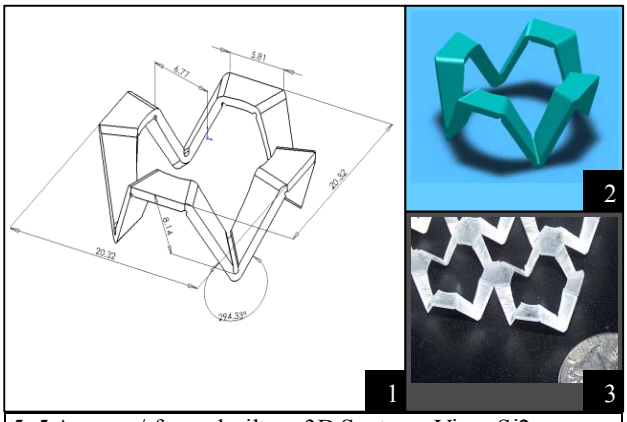
The first redesign involved modifying the leaf spring connectors of the unit cell. As depicted in Figure 27, the redesigned leaf spring only has two leaves versus the three leaf design produced by Becker to help eliminate the stiffness caused by the leaf spring connectors. Additionally, the new design employees a longer, thinner leaf than Becker's design.



5x5 Array w/ frame built on 3D Systems Viper Si2	
Mode: High Res	Room Temp: 77F
Line Width Compensation: Auto	Room Humidity: 26%
Z Correction: 1	Vat Temp: 25.4C
Resin: WaterClear 10120	Laser Power: 54mW

Figure 27 – First redesign of the unit cell that involves one less leaf and longer, thinner leaves for the leaf spring connectors

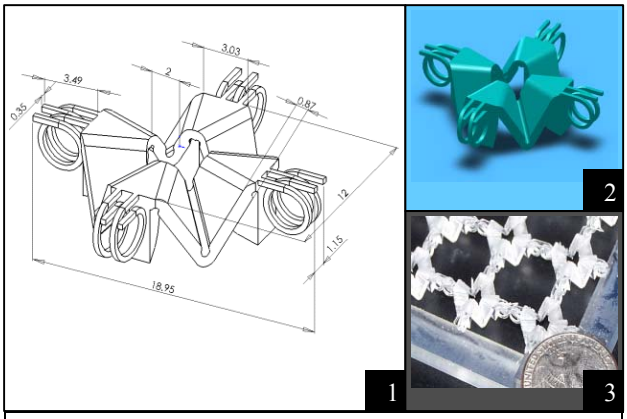
The next redesign strategy was to eliminate the leaf springs altogether, as shown in Figure 28. The size of the unit cell and the center hole were increased to produce a more elastic unit cell.



5x5 Array w/ frame built on 3D Systems Viper Si2	
Mode: High Res	Room Temp: 77F
Line Width Compensation: Auto	Room Humidity: 31%
Z Correction: 1	Vat Temp: 25.6C
Resin: WaterClear 10120	Laser Power: 59mW

Figure 28 – Illustration of a redesigned unit cell with an increased center hole and overall unit cell size to increase the elasticity of the unit cell

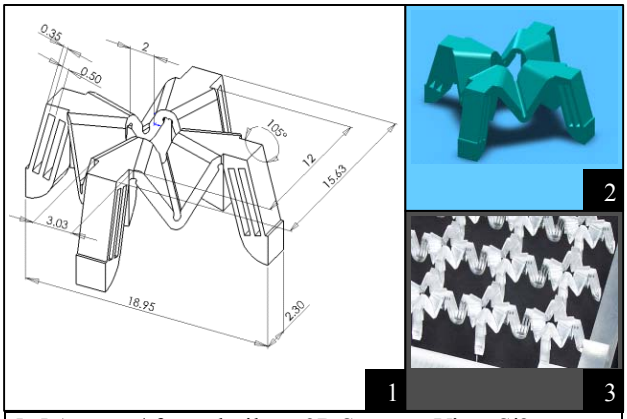
The third redesign disregarded the idea of using a leaf spring connector between unit cells to enable the surface to stretch. Instead, a helix was utilized in between the unit cells to allow the surface to stretch. The spherical compliant joint was kept the same as Becker's design. Figure 29 shows an illustration of the spherical compliant joint with helixes replacing the leaf springs of Becker's design.



5x5 Array w/ frame built on 3D Systems Viper Si2	
Mode: High Res	Room Temp: 77F
Line Width Compensation: Auto	Room Humidity: 33%
Z Correction: 1	Vat Temp: 25.9C
Resin: WaterClear 10120	Laser Power: 63mW

Figure 29 – Depiction of the unit cell redesign with helixes replacing the leaf springs of Becker's design

The last redesign of the unit cell that was created to relieve the compliant spherical joints of the stiffness produced by the leaf springs is shown in Figure 30. This unit cell was designed with the same dimensions as Becker's unit cell but the leaf springs were rotated by 90 degrees. The upper half of the rotated leaf spring was eliminated to produce the desired flat upper surface.



5x5 Array w/ frame built on 3D Systems Viper Si2	
Mode: High Res	Room Temp: 77F
Line Width Compensation: Auto	Room Humidity: 33%
Z Correction: 1	Vat Temp: 25.9C
Resin: WaterClear 10120	Laser Power: 63mW

Figure 30 – Redesigned unit cell with the leaf spring rotated 90 degrees and utilizing only half of the leaf spring

The abilities of these four additional redesigned unit cells to allow the compliant spherical joints to bend freely are evaluated in subsequent chapters.

The Mathematical Models

This section will analyze the kinematics developed for the square grid matrix that was discussed in previous sections. The kinematics equations provide a mathematical relationship between the positions of the actuators and the corresponding locations of the surface points of the mechanism. Both the forward and inverse kinematics are important for when the mechanism is used. The forward kinematics is used when the digital clay device is used as the input (when the physical space is being manipulated by the user), while the inverse kinematics is used when the digital clay device is used as the output (when the CAD drawing is being manipulated by the user). Knowing positions of the actuators, the forward kinematics hypothesizes where the position of the surface will be.

Some models for placing the surface points have been developed by Nguyen [11] and will be described further in this section. These methods of surface point placement are based on the links between the unit cells being rigid as shown in Figure 31.

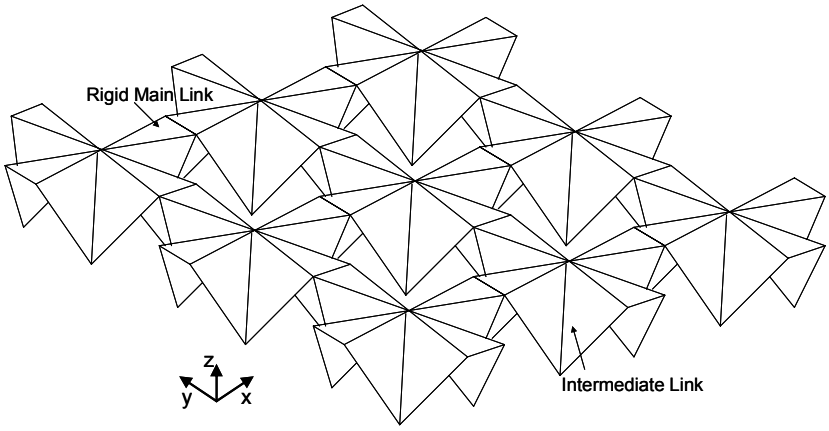


Figure 31 – Illustration of a 3-by-3 surface structure with rigid links

As one unit cell is actuated in the z-direction, representing the vertical movement of the surface, all connecting unit cells would actuate in the z-direction as while as shown in Figure 32. For the methods developed by Nguyen, inputs of z-coordinates of the unit cells are required. Since the x-coordinates and the y-coordinates are unknown values, the methods utilize an iterative process that maintains the length of the rigid links between the unit cells.

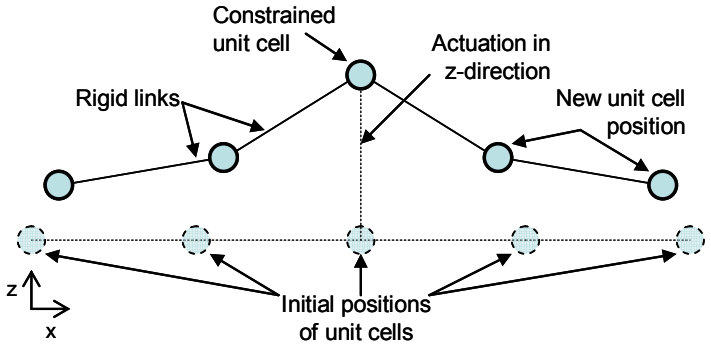


Figure 32 – Example of one unit cell being actuated and the reaction of the connected unit cells

The basic structure for Nguyen's mathematical methods is illustrated in Figure 33.

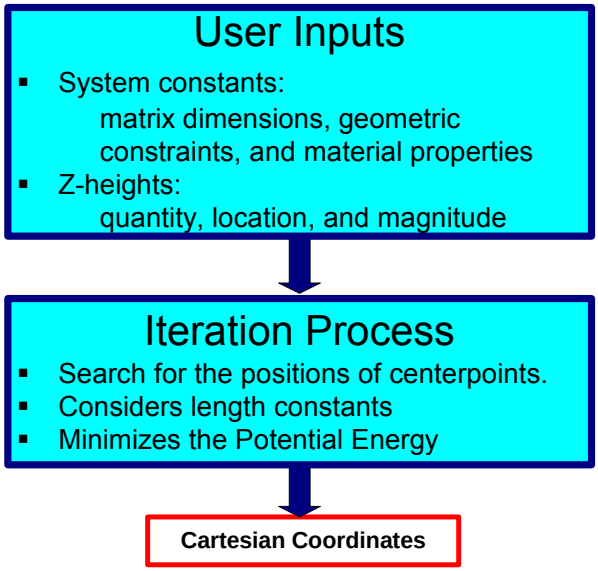


Figure 33 – System of steps for mathematical analysis procedure

The first mathematical model previously formulated by Nguyen to predict the movement and shape of the deformed surface required inputs for the desired z-coordinates, as mentioned previously, of the unit cells that were distorted. Some additional coordinate locations are required to be entered into the model. These additional coordinates represent the unit cells that are fixed in place on the crust. The deformed surface shape is found by a computational iteration of the desired unit cell z-coordinates, the fixed unit cell coordinates, and the distance constraint. The distance constraint is required in this mathematical model because the links connecting the unit cells are rigid.

The second formulation is very similar to the first mathematical model. The difference between the first mathematical model and the second mathematical model is the fact that the second model includes the use of an energy minimization objective. The joints of the spherical compliant unit cells are represented as theoretical rotational

springs, as shown in Figure 34. The rotational springs represent the stiffness of the compliant joints in the unit cells. This formulation was utilized to further develop mathematical models of the formable surface for this thesis. These additional mathematical models will be investigated later in further details.

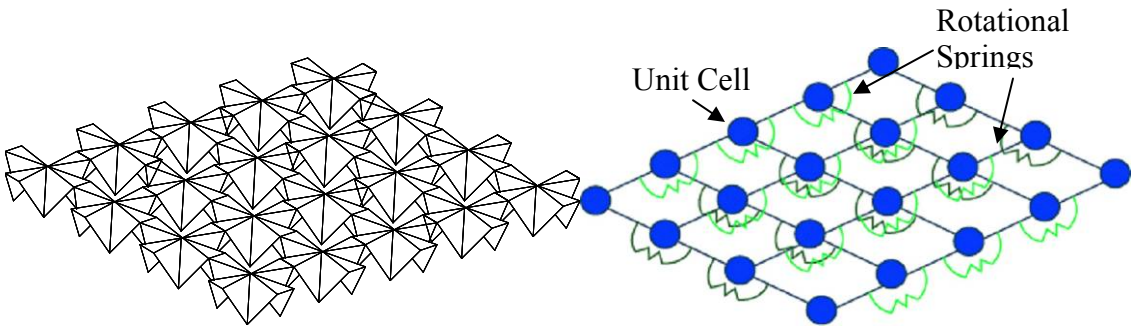


Figure 34 – An illustration of a 4-by-5 surface (left) and the spring representation of a 4-by-5 surface

A more detailed model is provided in the third and final formulation developed by Nguyen. This model illustrates, as shown in Figure 35, the fabricated surface design.

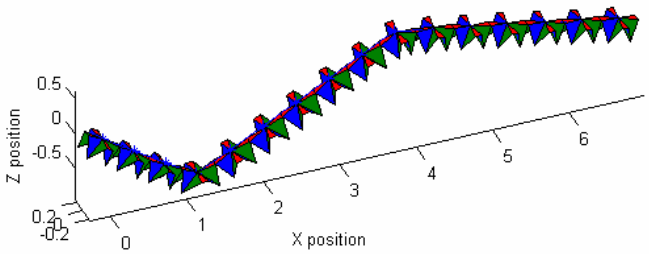


Figure 35 – Illustration of the third mathematical model developed my Nguyen

The position and orientations of links are given and the output is the joint angles for each of the four sides of the unit cell. Figure 36 has these joint angles labeled as θ_1, θ_2, and θ_3. Also illustrated in the figure are the direction vectors $\acute{o}$ and normal vectors $\acute{n}$, which are the inputs to the mathematical model.

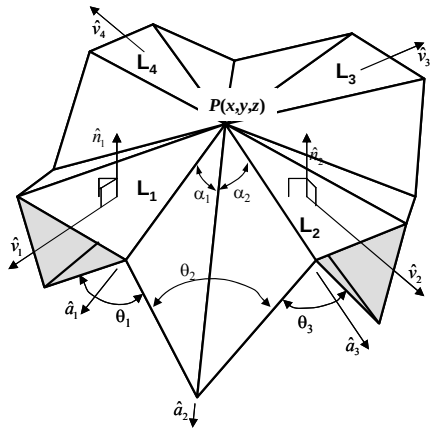


Figure 36 – Schematic of a spherical compliant joint unit cell

Nguyen created three basic formulation models followed by three solution algorithms to imitate digital clay. The first formulation utilized an iteration method that would output new unit cell positions from user defined system constraints and distance constraints between two unit cells. Unit cells were numbered using [row,column] notation, so that $\mathbf{P}_{i,j}$ refers to the coordinates of the unit cell center with indices [*i,j*].

Given:

m-by-*n* = size of matrix of unit cells center points in a uniform 2-D grid

$X_{f\,i,j}$, $Y_{f\,i,j}$, $Z_{f\,i,j}$ = fixed coordinate values of the [*i,j*] unit cell center points

dist = distance between two unit cell center points

k = average stiffness value of all the joints in the matrix

Find: $P_{i,j}$ = center points of every unit cell in the matrix

Satisfy: $\left\|P_{i+1,j} - P_{i,j}\right\| = dist$, $\left\|P_{i,j+1} - P_{i,j}\right\| = dist$ *(1)*

The first formulation developed by Nguyen is a nonlinear, underconstrained positioning problem. To solve this formulation, the fsolve function in MATLAB was used. The fsolve function finds a root of a system of nonlinear equations by utilizing the Levenberg-Marquardt iteration method [11].

The second formulation method applied a spring representation model where rotational springs were utilized to duplicate the stiffness of the spherical compliant joints.

This formulation is the same as the first formulation with the exception that this method adopted an iteration that outputted unit cell locations by minimizing the potential energy of the crust.

Minimize: $$P.E. = \frac{k}{2}\sum_{i=1}^{m*n}\left(\theta_{id} - \theta_{ir}\right)^2 \qquad (2)$$

where θ_{id} and θ_{ir} are the angles between a unit cell in the deformed and rest state.

Since the objective function for the second formulation is in the form of a sum of squares expression, Gauss-Newton and Levenberg-Marquardt algorithms were utilized to solve this formulation by applying the lsqnonlin function in MATLAB.

The third and final formulation utilized the full analytical model of the as manufactured unit cells. As with the second formulation, this formulation is similar to the first formulation with the exception of a complex minimization objective and the output values. The outputs in this formulation are the joint angles, shown in Figure 36, for each of the four sides of the unit cell.

Minimize: $$\frac{1}{2}\sum_{p=1}^{4*m*n}\left[k_1\left(\theta_{1p} - \theta_{1p}^*\right)^2 + k_2\left(\theta_{2p} - \theta_{2p}^*\right)^2 + k_1\left(\theta_{3p} - \theta_{3p}^*\right)^2\right] \qquad (3)$$

where k1 models the stiffness of joint 1 and joint 3 illustrated in Figure 37 and k2 models the stiffness of joint 2.

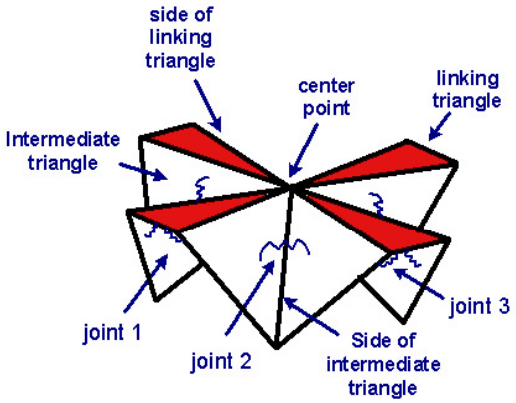


Figure 37 – Unit cell with labeled joints utilized in formulation 3

The third formulation is a highly dimensional nonlinear minimization problem that can be arranged in the same form as the second formulation. Because of this, an algorithm for nonlinear least squares minimization can be utilized.

The characteristics of the formulations were investigated by conducting two crust displacement tests [12]. The first test involved displacements that represent pushing up on the crust. The second test was conducted to depict a user dragging his or her finger across the surface of the crust. Each algorithm was tested to determine the total elapsed time for the algorithms to output the new unit cell positions. Table 1 lists the computational times for the three algorithms.

Table 1 – Computational times for the three developed algorithms [12]

	Algorithm 1		**Algorithm 2b**			**Algorithm 3**		
Crust Size	**Elapsed Time**	**Time per Iter.**	**Elapsed Time**	**Time per Iter.**		**Crust Size**	**Elapsed Time**	**Time per Iter.**
8x8	0.33217	0.04152	3.21725	0.65092		1x8	69.71	5.8092
10x10	0.85433	0.09493	9.07775	1.75720		1x10	93.785	9.3785
12x12	1.961	0.21789	23.6817	4.08537		1x12	130.418	13.0418
14x14	4.66483	0.46648	56.9067	8.57155		1x14	225.424	17.3403
16x16	8.9495	0.89495	111.681	16.4089		1x16	249.649	22.6954
18x18	18.2595	1.82595	236.332	31.0256		1x18	425.902	28.3935
20x20	32.969	2.99718	532.446	41.4942		1x20	395.419	35.9472
R^2 fit, 2nd order	0.9865	0.9922	0.9612	0.9938			0.987	0.9994
R^2 fit, 3rd order	0.9996	0.9991	0.9958	0.9939			0.9886	0.9999

By observing the computational times, it was concluded that the first and second algorithm methods out performed the third algorithm. The first two algorithms proved to be competitive in the accuracy of the output values even though they are less computationally involved than the third algorithm. The second algorithm produced smoother crust surface shapes than the first algorithm. Overall, the tests indicated that the second formulation method was the best out of the three methods. It portrays a reasonable manifestation of the crust in a moderate amount of time.

Auxiliary equations of the second formulation will be explored in the next chapter. These equations are based on the utilization of the spring representation that was exploited by Nguyen in the second formulation method. In addition to utilizing the spring representation, these auxiliary mathematical models will not incorporate the rigid link between unit cells. Instead, these models will include flexible links, between the unit cells, created by the leaf springs.

CHAPTER 5

REVISED MATHEMATICAL MODELS

Thus far a summary of previous research with a concentration on the surface of the digital clay device was established. In this chapter, additional derived mathematical models are explored. These auxiliary equations are based on the utilization of the spring representation that was exploited by Nguyen. This exploration will only examine the non-rigid surface for the digital clay device. The square crust pattern that was introduce in Chapter 4 can be easily modified to reflect the non-rigid surface by introducing prismatic joints, as shown in Figure 38. The prismatic joints for the manufactured crust were developed by Becker in the form of leaf springs [37].

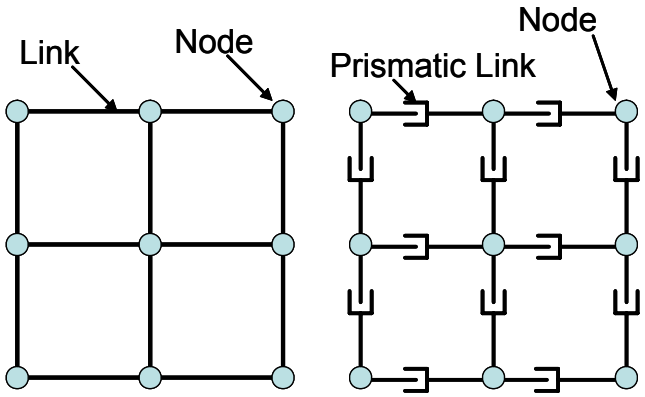


Figure 38 – Diagram of the square grid crust with rigid links (left), diagram of the square grid crust with prismatic joints that allows for stretching of the crust (right)

The majority of this chapter focuses on the two mathematical methods that were developed based on the utilization of the spring representation. For a more accurate representation of the leaf springs, translational springs were added between the unit cells in both mathematical models to represent the ability of the crust to stretch horizontally. Additionally, the spherical joints were represented as rotational springs in the two

mathematical models. An important concept applied in the second mathematical model was that a prismatic joint permits relative translational movement between two bodies, in this case unit cells, but no relative rotational movement at all. This physical concept was imitated by introducing additional rotational springs to the second mathematical model.

The next section in this chapter briefly explores the concepts involved with the spring representation, which is utilized in the mathematical models that are discussed later in this chapter. The spring representation involves depicting the spherical joints as rotational springs and designating the leaf springs as translational springs.

The Objective Function

The spring representation method for modeling the surface of the digital clay device is based on the relative range of motion of the unit cells. This model indicates that the angle of rotation across unit cells, θ, can deviate by a nominal angle, θ^*, and that the distance, x, between each unit cell can vary by a nominal displacement, x^*. The components of the unit cells of the surface try to minimize the variation between a stretched surface and a surface at rest. This is achieved by the spherical joints minimizing the angular displacements and the leaf springs minimizing the displacements in length. An objective function, f, can be created to achieve this minimization:

$$f(x, \theta) = f((x_0 - x_0^*, \ldots, x_m - x_m^*), (\theta_0 - \theta_0^*), \ldots, (\theta_n - \theta_n^*)) \quad (4)$$

where there are m number of links and n number of angles between contiguous links. The objective function, in this case f, is the function to be optimized. In this case the input parameters are the actuator heights that cause variation between the angle/lengths and the nominal displacements amid unit cells. If a set of unit cell positions is given by P, and the objective function is expressed as $f(P)$, then the desired deformed unit cell positions, P_d, satisfy:

$$f(P_d) = \min_{P \in S} f(P) \quad (5)$$

$$S = \{P \mid h(P) = 0\} \quad (6)$$

where the constraints that all unit cells must lie on the deformed surface and that a minimum of two unit cells are constrained. The expression $h(P) = 0$ is a representation of the unit cells that are fully constrained. One constraint for unit cells is to have a fixed point that prevents the surface from free floating in space. The second constraint is used to prevent the surface from inverting around the fixed point.

Spring Model to the Mathematical Model

The objective function described can be used with the spring representation in a mathematical model. The spring representation replaces the leaf springs with translational springs and adds rotational springs to represent the bending across the spherical joints. The leaf springs were introduced in Chapter 4 in the section titled Leaf Spring Connections. Additionally, the original Figure 34 in Chapter 4 illustrates a spring representation of the rigid surface with just the rotational springs. Figure 39 shows a CAD model of the non-rigid surface and two spring representation models that will be discussed later in more detail.

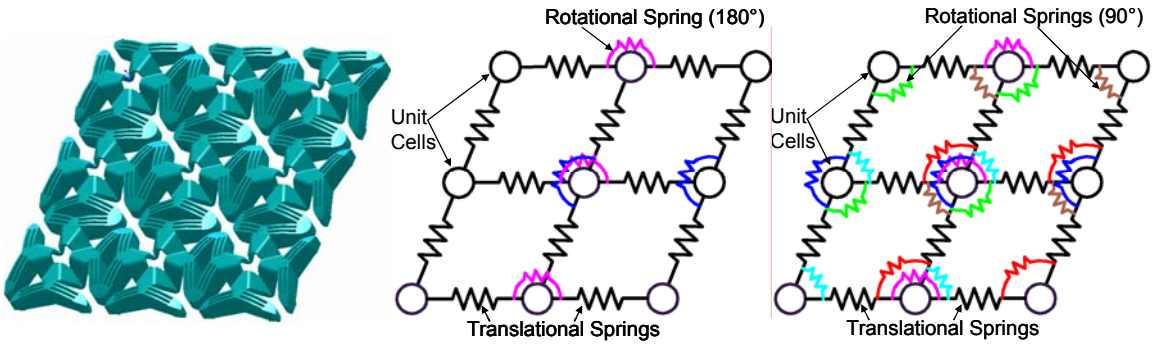


Figure 39 – CAD model of non-rigid surface (left), spring representation with translational and 180 degree rotational springs (middle), spring representation with translational, 90 degree rotational springs, and 180 degree rotational springs (right)

In the spring representation models, the objective function f is the minimization of the total potential energy of the crust. The output results in unit cell positions where the entire surface is in static equilibrium. The solution, P_d, minimizes the total potential

energy stored in the springs, *f*, to obtain a state of static equilibrium. The state of static equilibrium is obtained when each unit cell has a sum of forces equal to zero. An advantage to the spring representation is the results are physically meaningful. The physical system has joints with bounded stiffness; the surface will conform to a natural state even if only a few joint coordinates are fixed.

Since the non-rigid links were represented as translational springs, a linear stiffness k_{trans} was assigned to it. It also follows that an angular stiffness k_{rotat} was assigned to the rotational springs that represent the stiffness of the spherical joint. The general objective function, *f*, now becomes:

$$f = \frac{1}{2} k_{trans} \sum_{i=1}^{c} \left(x_i - x_i^* \right)^2 + \frac{1}{2} k_{rotat} \sum_{j=1}^{g} \left(\theta_j - \theta_j^* \right)^2 \qquad (7)$$

where *c* is the number of translational springs and *g* is the number of rotational springs. Given an *m*-by-*n* crust matrix, as labeled in Figure 40, the number of translational springs can be found by:

$$c = m(n + 1) + n(m - 1) \qquad (8)$$

And the number of rotational springs can be found by:

$$g = m(n - 2) + n(m - 2) \qquad (9)$$

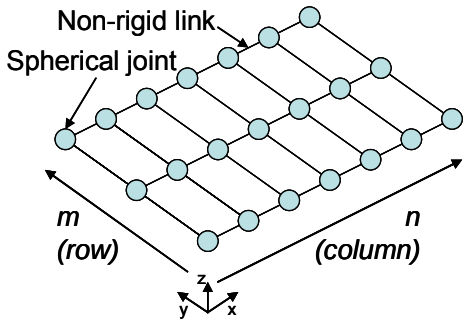


Figure 40 – Example of an m-by-n (3-by-7) matrix showing the row-column notation

Many different approaches to minimize the potential energy of the crust could be utilized. The technique that is presented in this thesis involves an all-inclusive approach

in which a single optimization of the total potential energy of the surface will be performed. Using equation *(7)* to find the potential energy per unit cell of an m-by-n surface would result as:

$$P.E. = \frac{1}{2}\sum_{p=1}^{m*n}\left[k_{trans}\left(x_p - x_p^*\right)^2 + k_{rotat}\left(\theta_p - \theta_p^*\right)^2\right] \qquad (10)$$

where P is the node or unit cell, m is the number of rows, and n is the number of columns. The objective function in equation *(10)* is in the form consisting of a sum of squares expression. The form of a sum of squares is the same form as a nonlinear least squares problem:

$$\min f(x) = \sum_i f(x_i)^2 \qquad (11)$$

As shown in equation *(11)*, least squares is a mathematical optimization technique that attempts to find a function which closely approximates a given series of measured data into a best fit. It attempts to minimize the sum of the squares of the residuals between points generated by the function and corresponding points in the data.

One important issue of the formable crust is the fact that each unit cell has one degree-of-freedom per compliant joint and one degree-of-freedom per leaf spring. The number of equations is less than the number of unknowns which makes this system underconstrained. One way to resolve this issue is to have the number of constraints equal the number of unknowns but since the compliant joints have a restricted stiffness the surface will take an appropriate form even if a minuscule number of constraints are present. This alludes to what types of mathematical models can be implemented. Since the objective function has the same form as a nonlinear least-squares problem, Gauss-Newton and Levenberg-Marquardt algorithms can be utilized in the mathematical models [39].

The mathematical models described in the following sections are intended to imitate user manipulations in real-time response. The user inputs the desired size of the

m-by-n crust matrix, the desired z-coordinates of the unit cells in regards to the deformed crust, and the coordinates of the fixed unit cells. The movements of the fabricated crust are generated by the displacement of the actuators. The mathematical models output the cartesian coordinates of the unit cells that produces the minimal amount of potential energy that satisfies the constraints implemented on the crust.

First Mathematical Model

The objective of the first mathematical model was to establish a crust shape that satisfied the desired unit cell z-coordinates and fixed unit cell coordinates, while minimizing the potential energy of the crust. This model used a conceptual representation where the circles in Figure 41 represent the center points of the unit cells in a 3-by-3 matrix and the jagged lines connecting the circles represent the non-rigid links between the unit cells that are created by the leaf springs.

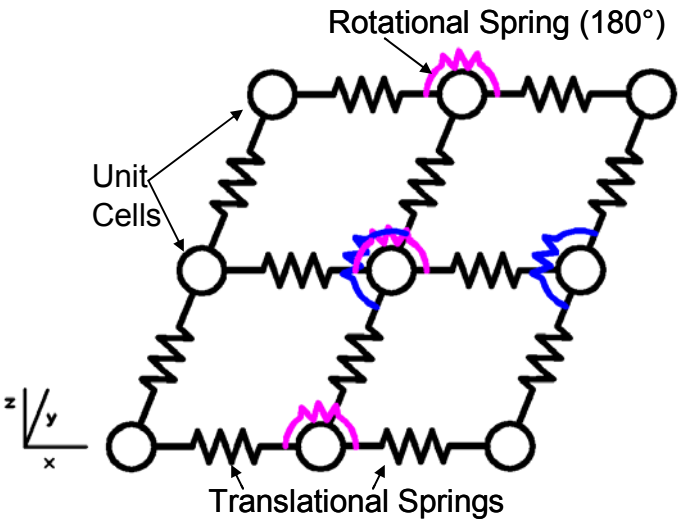


Figure 41 – A graphical illustration of the spring representation used in the first algorithm; this spring representation utilizes two different spring types: translational springs, and 180 degree rotational springs

Translational springs were used to model the stiffness of the leaf springs in the unit cells. As the surface deforms, the unit cells move and the ends of the links are free

to pivot about the center points of the unit cells. Rotational springs with 180 degree offsets, shown in Figure 42, were used to model the stiffness of the compliant joints in the unit cells.

Figure 42 – Illustration of a rotational spring with 180 degree offset

Keeping the same formulation notation as previous research [11, 12], unit cells are numbered using [row,column] notation, so that $\mathbf{P}_{i,j}$ refers to the coordinates of the unit cell center with indices [*i*,*j*]. In the following mathematical formulation, the desired position of the unit cells, $\mathbf{P}_{i,j}$, is such that the potential energy, *PE*, of an *m*-by-*n* surface is minimized.

Given:

m-by-*n* = size of matrix of unit cells

$x_{fi,j}$, $y_{fi,j}$, $z_{fi,j}$ = fixed coordinate values of [*i*,*j*] cell center points

Find: $\mathbf{P}_{i,j}$ = center points of every unit cell in the matrix

Minimize:
$$PE = k_{trans}\frac{1}{2}\sum_{i=1}^{c}(d_{id}-d_{ir})^2 + k_{rotat}\frac{1}{2}\sum_{i=1}^{g}(\theta_{id}-\theta_{ir})^2 \qquad (12)$$

where k_{trans} and k_{rotat} are the spring constants, *c* is the number of translational springs given in equation *(8)*, *g* is the number of rotational springs given in equation *(9)*, θ_{id} and θ_{ir} are the angles of rotation across a unit cell in the deformed and rest positions, d_{id} and d_{ir} are the distances between two connected unit cells in the deformed and rest positions. Figure 43 illustrates the angles θ_{id} and θ_{ir} and the distances d_{id} and d_{ir} along a cross sectional view on the x-z plane. In this illustration, the middle node, or unit cell, was

actuated vertically in the z-direction and the remaining unit cells are in their initial positions.

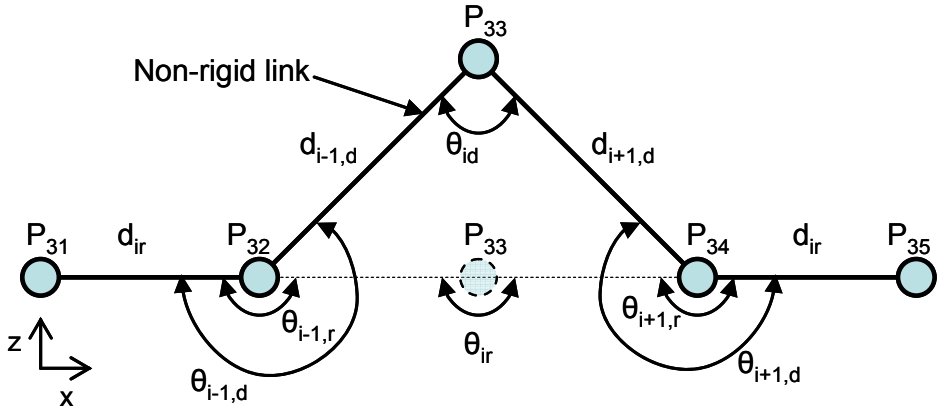


Figure 43 - Cross-sectional schematic of the clay illustrating angle orientation

As can be seen in Figure 43, angle θ_{ir} was and always is equal to π. Angle θ_{id} was calculated by computing the angle that opposing links make across the unit cell $P_{i,j}$. The equation for determining θ_{id} is:

$$\theta_{id} = \tan^{-1}\left(\frac{P^z_{i_1,j_1} - P^z_{i,j}}{P^h_{i_1,j_1} - P^h_{i,j}}\right) - \tan^{-1}\left(\frac{P^z_{i,j} - P^z_{i_1,j_1}}{P^h_{i,j} - P^h_{i_1,j_1}}\right) \qquad (13)$$

where $P^z_{i,j}$ equals the z-coordinate value of the unit cell $P_{i,j}$, and $P^h_{i,j}$ equals the h-coordinate value of the unit cell. The h-coordinate is a representation of either the x-coordinate or the y-coordinate depending on the direction of the links being investigated. Additionally, the values of h, i_1, and j_1 depend on the direction of the links being investigated. If the x-direction is being investigated, then h is x, i_1 is i, and j_1 is $j+1$. On the other hand, if the y-direction is being investigated, then h is y, i_1 is $i+1$, and j_1 is j. For instance, the investigated links in the x-direction across the unit cell labeled $P_{i,j}$ in Figure 44 are the link connecting $P_{i,j}$ to $P_{i+1,j}$ and the link connecting $P_{i,j}$ to $P_{i-1,j}$. For the y-direction, the examined links are the link connecting $P_{i,j}$ to $P_{i,j+1}$ and the link connecting $P_{i,j}$ to $P_{i,j-1}$.

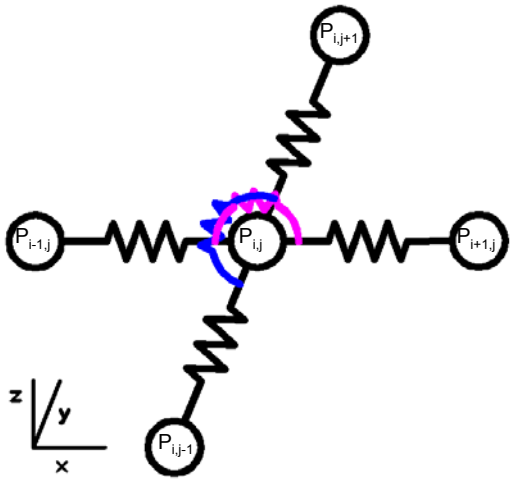


Figure 44 – Spring representation with labeled nodes to assist in investigating the links utilized determining θ_{id} shown in Figure 43

A closer investigation of Figure 43 will reveal the value of d_{ir} is always the original distance between unit cells, which is 0.8 inches or 20.32 mm in this case. The numerical value of the original distance was determined from the given distance between each actuator pin in the bed-of-nails system platform. Distance d_{id} is calculated by using the Pythagorean theorem. The equation for determining d_{id} is:

$$d_{id} = \sqrt{\left(P^{h}_{i_1,j_1} - P^{h}_{i,j}\right)^2 - \left(P^{z}_{i_1,j_1} - P^{z}_{i,j}\right)^2} \qquad (14)$$

where $P^{z}_{i,j}$ equals the z-coordinate value of the unit cell $P_{i,j}$, and $P^{h}_{i,j}$ equals the h-coordinate value of the unit cell. As with the angle θ_{id}, the value of h, i_1, and j_1 depends on the direction of the springs being investigated. If the springs in the x-direction are being investigated, then h is x, i_1 is i, and j_1 is $j+1$. On the other hand, if the springs in the y-direction are being investigated, then h is y, i_1 is $i+1$, and j_1 is j. For instance, the investigated springs in the x-direction in Figure 45 are labeled Spring_1 and Spring_2. The value of d_{id} for Spring_1 will equal d_{ir} because the spring is not stretched. However, the value of d_{id} for Spring_2 can be determined by using equation *(14)*. The z-coordinate value of unit cell $P_{i+1,j}$ is determined by the displacement of the actuator.

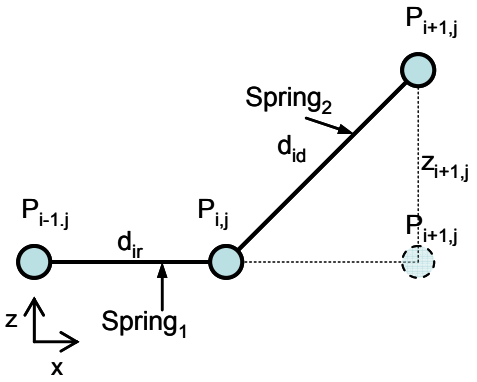


Figure 45 - Example for determining the value of d_{id}

Second Mathematical Model

The second mathematical model developed utilizes equation *(12)* and is similar to the previously described model. The difference between the two mathematical models is the type and number of rotational springs that are used in the representations. The first mathematical model utilized translational springs and rotational springs with an offset of 180 degree. The second mathematical model applies both springs utilized in the first mathematical model with an addition quantity of rotational springs with an offset of 90 degrees. Equation *(8)* and equation *(9)* are still employed to calculate the number of translational springs and rotational springs with an offset of 180 degree but equation *(15)* is appropriate to determine the number of rotational springs with an offset of 90 degree:

$$o = 4(n - 1)(m - 1) \qquad (15)$$

The adaptation of the additional rotational springs is incorporated to mimic the removal of relative rotational movement caused by the leaf spring. A comparison of the first and second representation is shown in Figure 46.

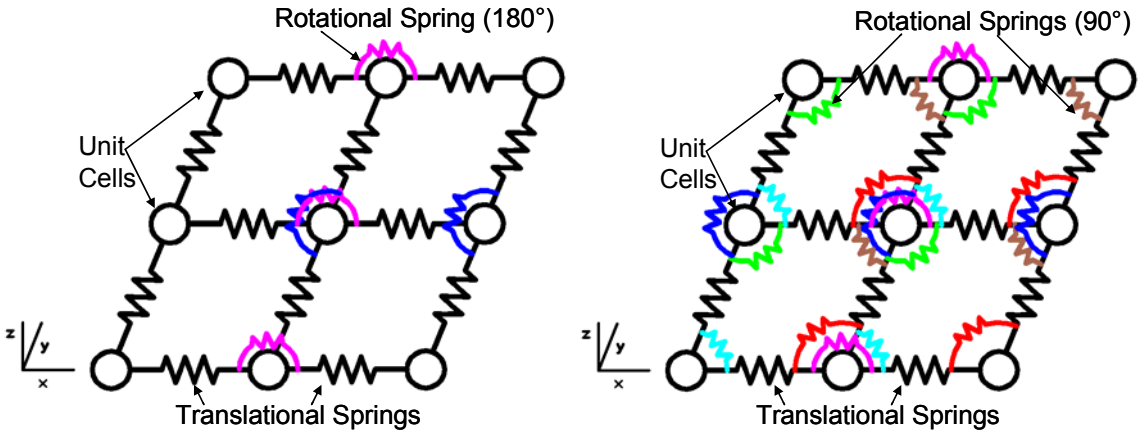


Figure 46 - First spring representation (left), second spring representation (right)

Because of the increased number of rotational springs employed in the second mathematical model, the minimization term utilized in the first mathematical model is slightly modified.

Find: $\mathbf{P}_{i,j}$ = center points of every unit cell in the matrix

Minimize: $$PE = k_{trans}\frac{1}{2}\sum_{i=1}^{c}(d_{id}-d_{ir})^2 + k_{rotat}\left(\frac{1}{2}\sum_{i=1}^{g}(\theta_{id}-\theta_{ir})^2 + \frac{1}{2}\sum_{i=1}^{o}(\alpha_{id}-\alpha_{ir})^2\right) \quad (16)$$

where o is the number of rotational springs with an offset of 90 degrees given in equation *(15)*, α_{id} and α_{ir} are the angles of perpendicular rotation across a unit cell in the deformed and rest positions.

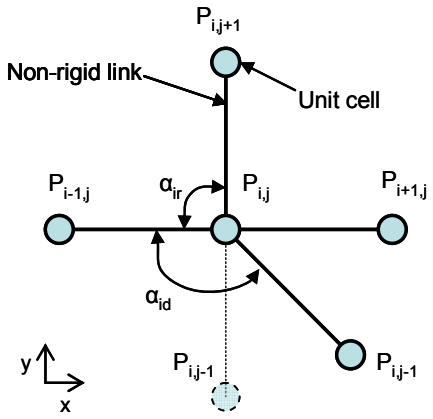


Figure 47 – Illustration showing links associated with determining α_{id}

As can be seen in Figure 47, angle α_{ir} is always equal to $\pi/2$. Angle α_{id} is calculated by computing the angle that intersecting links make across the unit cell $P_{i,j}$. The equation for determining α_{id} is:

$$\alpha_{id} = \tan^{-1}\left(\frac{P^y_{i,j_1} - P^y_{i,j_2}}{P^x_{i,j_1} - P^x_{i,j_2}}\right) - \tan^{-1}\left(\frac{P^x_{i_1,j} - P^x_{i_2,j}}{P^h_{i_1,j} - P^h_{i_2,j}}\right) \qquad (17)$$

where i_1, i_2, j_1 and j_2 depend on which unit cells are being investigated. For example, the dashed unit cells illustrated in Figure 48 signifies the values of i_1, i_2, j_1 and j_2 to be $i+1$, i, $j+1$, and j respectively.

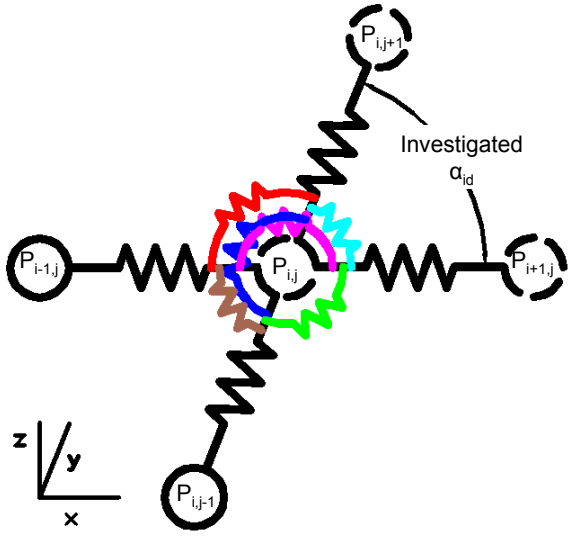


Figure 48 –Example for determining angle α_{id}

Additional Application to the Mathematical Models

The mathematical models described in this chapter have objective functions that were utilized to establish a crust shape that satisfies unit cell constraints while minimizing the potential energy of the crust. A similar approach can be applied to minimize the error between a desired shape and the simulated surface shape obtained by utilizing the mathematical models. In the following error reduction formulation, the same

notation that was used previously was applied. Furthermore, the desired position of the unit cells attached to an actuator, $\boldsymbol{P}_{i,j}$, is such that the error between the desired surface shape and the simulated surface shape of an *m*-by-*n* surface is minimized.

Given:

m-by-*n* = size of matrix of unit cells

$\boldsymbol{C}_{i,j}$ = calculated position of every unit cell utilizing mathematical model

Find: $\mathbf{P}_{i,j}$ = position of every unit cell attached to an actuator

Minimize: $Error = \sum f(D_{i,j} - C_{i,j})$ *(18)*

where $\boldsymbol{D}_{i,j}$ is the position of every unit cell of the desired surface shape. Equation (18) can be put into the form of a sum of squares problem as shown in Equation (19) and Equation (20).

$$Error = \sqrt{\sum f(D_{i,j} - C_{i,j})^2} \quad (19)$$

$$Error^2 = \sum f(D_{i,j} - C_{i,j})^2 \quad (20)$$

Additional methods for solving the error minimization function are possible but were not explored in this thesis. Figure 49 illustrates the error minimization of the formulation previously describes. As shown in the figure, the displacement of the unit cells, $\boldsymbol{P}_{i,j}$, attached to the actuators can be utilized to decrease the error between calculated unit cell positions, $\boldsymbol{C}_{i,j}$, and the desired unit cell positions, $\boldsymbol{D}_{i,j}$.

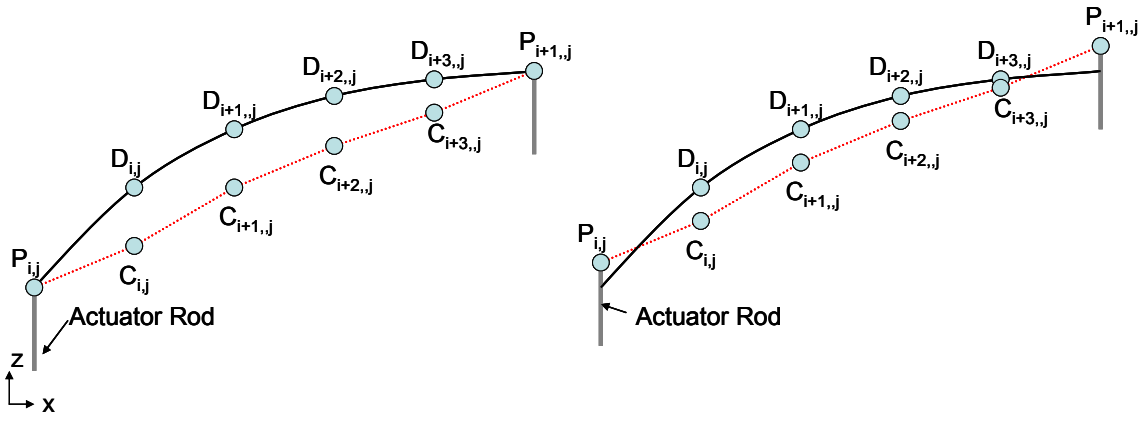


Figure 49 – Initial actuator position (left) and displacing the actuators to decrease the error between desired unit cell locations and calculated unit cell locations (right)

Review of the Mathematical Models

In this chapter two mathematical models were introduced and discussed. Both mathematical models utilized a spring representation that consisted of two different set of springs to mimic the behavior of the leaf spring connections and the spherical joints of the unit cells. Translational springs were applied in the two mathematical models to mimic the ability of the crust to stretch horizontally. The second set of springs employed in the mathematical models was rotational springs. The rotational springs model the stiffness of the compliant spherical joints in the crust.

Now that some mathematical models have been defined, an algorithm for finding the minimum potential energy and the new unit cell positions of the crust was needed. The next chapter introduces and discusses the algorithms involved for solving the mathematical models.

CHAPTER 6

SOLUTION ALGORITHMS

Minimization algorithms were needed to determine the unit cell positions for the non-linear, multidimensional, under constrained mathematical models presented in the previous chapter. The major difference between linear and non-linear models is that model parameters need to be estimated for a nonlinear mathematical model. In the case of the two mathematical models previously analyzed, the initial parameters were the coordinates of the actuators and the coordinates of other fixed unit cells.

As mentioned in Chapter 5, the objective function of the mathematical models was to minimize the totality of the potential energy in the crust. The two following equations are in the form of a nonlinear least squares formulation and represent the objective functions of the mathematical models:

$$PE = k_{trans}\frac{1}{2}\sum_{i=1}^{c}(d_{id}-d_{ir})^2 + k_{rotat}\frac{1}{2}\sum_{i=1}^{g}(\theta_{id}-\theta_{ir})^2 \qquad (21)$$

$$PE = k_{trans}\frac{1}{2}\sum_{i=1}^{c}(d_{id}-d_{ir})^2 + k_{rotat}\left(\frac{1}{2}\sum_{i=1}^{g}(\theta_{id}-\theta_{ir})^2 + \frac{1}{2}\sum_{i=1}^{o}(\alpha_{id}-\alpha_{ir})^2\right) \qquad (22)$$

MATLAB's Optimization Toolbox was used to solve the minimization of equation *(21)* and equation *(22)* utilizing the MATLAB function lsqnonlin. This MATLAB function was invoked as:

options = optimset ('Display','iter', 'TolFun', 0.002, 'TolX', 0.002);

[x,resnorm,fval,exitflag,output] = lsqnonlin(ffun,xvec,options)

MATLAB utilizes the Levenberg-Marquardt method with a line search algorithm that is a mixed quadratic and cubic polynomial interpolation and extrapolation method [40]. Even though the Levenberg-Marquardt algorithm tends to be slower than the Gauss-Newton algorithm, Levenberg-Marquardt is more robust. This means that in many cases it finds a

solution even if the initial guesses are far from the final position. The MATLAB program in its entirety can be viewed in Appendix A.

Algorithm Experiments

Two types of experiments were constructed to examine the disposition of the two algorithms. These algorithms do not take into account the effect of gravity acting upon a fabricated surface or the effect of friction caused by a user dragging his or her finger across the crust. Algorithm 1 is the algorithm that was utilized to calculate the unit cell positions of the first mathematical model. Furthermore, Algorithm 2 is applied for solving the second mathematical with the additional rotation springs. An arbitrary translational spring constant value was utilized in both experiments while an average value of Nguyen's spring constant was used for the rotational spring [11]. The spring constant value of 1900 N/m was used for the leaf spring while 0.0756 N·m was applied to the spherical joint. The first experiments represented a user pushing up on the center of the crust. The intent of this investigation was to illustrate if the algorithms produce reasonable solutions, such as a symmetric form of the crust. The purpose of the second investigation of the algorithms was to test the response time of a more complex manipulation of the crust. This manipulation mimicked a user pushing against the crust and dragging his or her finger across the surface. These experiments were conducted on a Dell Dimension 8400 Workstation with an Intel Pentium 4 3.20 GHz processor and 2 GB of RAM.

Push Experiment

For this experiment, the size of the test crust was increased by increments of 2-by-2. The smallest array of unit cells tested was 5-by-5 and the largest was 21-by-21. The unit cells located at the four corners of the crust were fully constrained in the x, y, and z-directions, as illustrated in Figure 50. To replicate the motion of a user pushing up on the

crust, the unit cell located at the center of the array incurred five height displacements in increments of 2.5 mm [12]. It was assumed that the user only moved his or her finger straight up into the crust, so the center unit cell was constrained in the x-direction and y-direction.

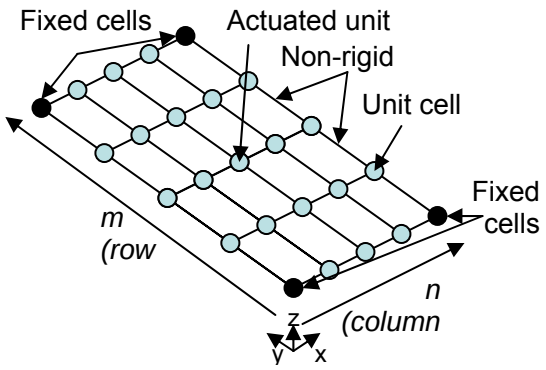


Figure 50 – Schematic of a 5-by-5 crust with constrained corner unit cells

The general shape of the crust generated by both algorithms resulted in shapes that resembled a cone. These shapes seemed reasonable because the links between the unit cells were slightly stretched, not compressed. Figure 51 illustrates the results of Algorithm 2, where the green-dashed lines are the initial positions and the solid-blue lines are the positions after the user pushed up on the crust. The shape of the crust produced by Algorithm 2 was smoother than the crust produced by Algorithm 1.

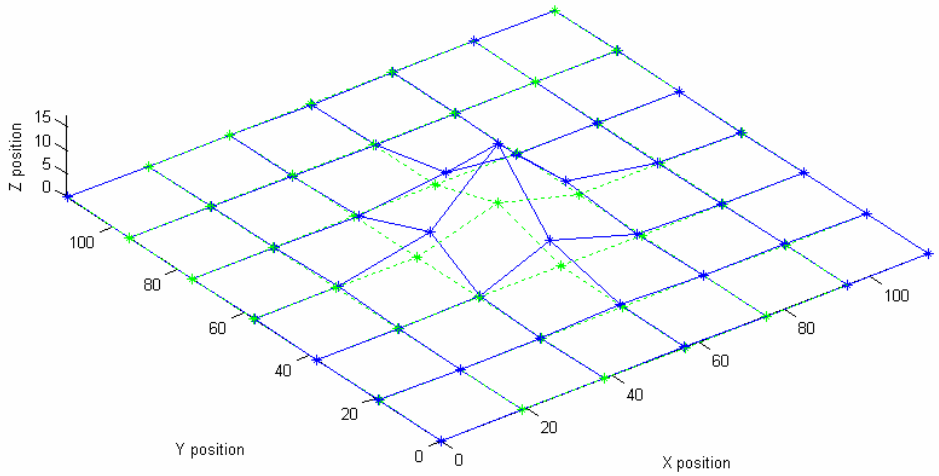


Figure 51 – Results from Algorithm 2 of the push experiment on a 7-by-7 array

The average computation times, listed in seconds, for the two algorithms are shown in Table 2. Both algorithms converged relatively quickly for crust sizes smaller than 15-by-15 but the convergence time doubles for the 15-by-15 array. Algorithm 2 was expected to converge to a solution slower than Algorithm 1 because it incurs a more intense iteration because of the additional rotational springs. The average time per iteration illustrates the computational characteristics of the algorithms without taking into consideration the number of total iterations needed. The nature of the algorithm can be better represented by this because it was independent of the iteration size, which can vary.

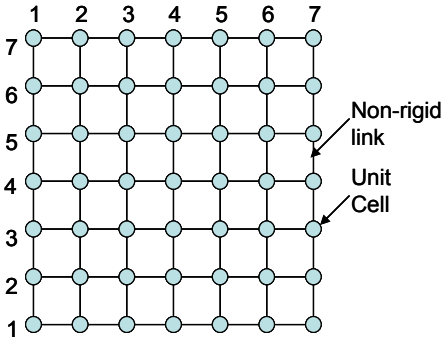
The last column shown in Table 2 labeled "X & Y Boundary Condition" shows the average computation times for both algorithms with all x-coordinates and y-coordinates constrained. A minuscule difference between the computational times of Algorithm 1 and Algorithm 2 was produced by applying the x and y constraints on both algorithms, so an average was taken to produce the data in Table 2. By adding more constraints to the system, the computational time decreased.

Table 2 – Average computation times for the two algorithms during the push experiment

	Algorithm 1		Algorithm 2		X & Y Boundary Condition	
Crust Size	**Time per iteration (sec)**	**Elapsed Time (sec)**	**Time per iteration (sec)**	**Elapsed Time (sec)**	**Time per iteration (sec)**	**Elapsed Time (sec)**
5-by-5	0.023701	0.939582	0.025447	1.006144	0.017835	0.442135
7-by-7	0.051572	2.314484	0.062694	2.385846	0.027274	1.226891
9-by-9	0.130741	4.173462	0.142062	5.504619	0.050723	3.765371
11-by-11	0.266827	10.257006	0.343877	13.620091	0.091964	8.945307
13-by-13	0.410706	15.881151	0.551736	21.954834	0.163112	15.148135
15-by-15	0.890928	34.761982	1.151587	46.671761	0.283478	28.064289
17-by-17	1.484391	58.043756	1.908403	79.008217	0.462461	45.783624
19-by-19	2.366287	93.531594	2.533629	104.996229	0.72045	71.324599
21-by-21	3.640742	143.840174	4.561330	187.057423	1.107994	109.6914
R^2, 2nd order	0.9995	0.9996	0.9905	0.9914	0.9995	0.9996
R^2, 3rd order	0.9995	0.9996	0.993	0.9933	0.9995	0.9996

The coefficient of determination, R^2, values for least-squares regressions are shown in Table 2. R^2 is used to determine how well a regression fit applies to the data set. For this experiment, it represents the fraction of variability in the amount of time elapsed that can be explained by the variability in the number of total unit cells of the tested crust. The higher the R^2 value is, the more useful the model becomes. The data in Table 2 was fitted to second and third degree polynomials to determine the complexity of the algorithm. The R^2 values for both Algorithm 1 and Algorithm 2 are high but the differences between the second order and third order R^2 values are minuscule, so no conclusion can be determined.

Since the crust is being actuated in the center of the crust and the four corner unit cells are fixed, the results should be relatively symmetric. The z-values of a 7-by-7 array of unit cells used to analyze if the values are symmetric are shown in Figure 52. The results shown in Figure 52 indicated that the resulting crust shape was not symmetric but the problem was symmetric.



	1	2	3	4	5	6	7
7	0	-2.37E-05	1.421E-05	0.001045	-1E-06	-8.1E-06	0
6	-2.37E-05	2.489E-05	0.0004351	0.1823	0.000598	-3.21E-05	3.119E-06
5	1.422E-05	0.0004351	-0.004082	4.3094	0.15628	0.0012195	-3.96E-05
4	0.0010448	0.1823	4.3099	17.5	3.8938	0.15758	0.0014556
3	-1.02E-06	0.000598	0.15628	3.8938	0.078387	9.89E-05	-1.63E-05
2	-8.1E-06	-3.21E-05	0.0012342	0.15756	9.9E-05	-3.69E-05	1.76E-05
1	0	3.121E-06	-3.96E-05	0.001456	-1.6E-05	1.76E-05	0

Figure 52 – Schematic of a 7-by-7 crust (left), related z-values of a 7-by-7 crust actuated in the center

One reason for the non-symmetric results shown in Figure 52 is the possibility that the mathematical model was computing local minima from the initial guess to create the least potential energy of the surface as a whole. Figure 53 illustrates how two different unit cell locations produce the same local minimum potential energy.

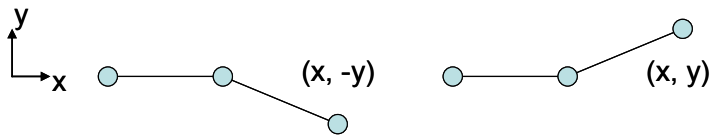
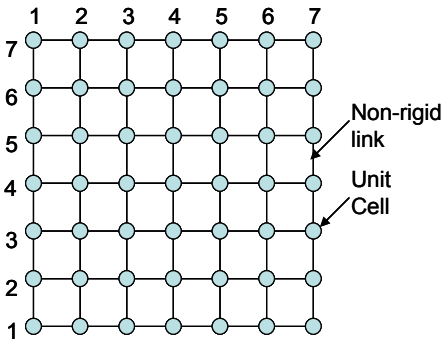


Figure 53 – Illustration of two different unit cell locations that result in the same potential energy

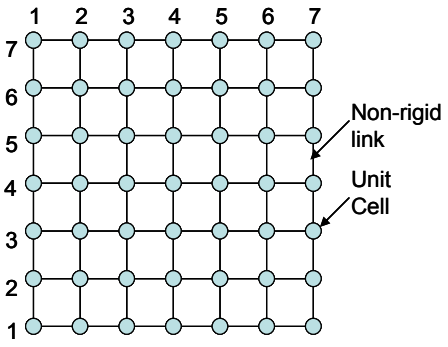
To evaluate this hypothesis, two tests were imposed on the mathematical model. The first test consisted of an evaluation of the z-values of a 7-by-7 array of unit cells with constraints on all x-coordinates and y-coordinates. The results shown in Figure 54 are symmetric. This indicates that additional constraints force the mathematical model into producing a unique and symmetric solution because the same z-values were produced after multiple iterations of the mathematical model.



	1	2	3	4	5	6	7
7	0	3.7792	6.3511	7.7344	6.3511	3.7792	0
6	3.7792	5.9754	8.4861	9.9585	8.4861	5.9754	3.7792
5	6.3511	8.4861	11.238	13.327	11.238	8.4861	6.3511
4	7.7344	9.9585	13.327	17.5	13.327	9.9585	7.7344
3	6.3511	8.4861	11.238	13.327	11.238	8.4861	6.3511
2	3.7792	5.9754	8.4861	9.9585	8.4861	5.9754	3.7792
1	0	3.7792	6.3511	7.7344	6.3511	3.7792	0

Figure 54 – Schematic of a 7-by-7 crust (left), related z-values of a 7-by-7 crust actuated in the center with constraints inflicted on the x and y-coordinates

The second test utilized different initial values for the z-values. During this test, the initial guesses of the z-values were set in such a way that it resembled a square pyramid. The system constraints were kept the same during this test as the previous test. The results are shown in Figure 55. These results seemed more reasonable than the results shown in Figure 52 because there were no negative valued z-coordinates and the z-values appeared more symmetric. A comparison between the residual amounts obtained utilizing the initial guess of zero for all of the z-values and the pyramid values for the initial guess of the z-values was also completed. The residual amount for the zero z-value initial condition was 9434.17. The residual assessment for the pyramid configuration was remarkably less with an amount of 0.346. This is a good indication that the zero z-value initial condition is not a good guess and that the output of the position locations by the mathematical model is dependent on the initial guess of the positions.



	1	2	3	4	5	6	7
1	0	4.0753	6.9192	7.9463	6.9127	4.0759	0
2	4.0674	5.7985	8.1045	9.6209	8.1357	5.8079	4.0707
3	6.9169	8.1341	11.18	13.836	11.114	8.1372	6.8934
4	7.9468	9.6195	13.831	17.5	13.872	9.6125	7.9477
5	6.918	8.1398	11.084	13.873	11.037	8.141	6.9007
6	4.0664	5.8015	8.1559	9.6167	8.1354	5.7919	4.0631
7	0	4.0842	6.8924	7.9479	6.902	4.0835	0

Figure 55 – Schematic of a 7-by-7 crust (left), related z-values of a 7-by-7 crust actuated in the center with initial z-coordinates in the shape of a square pyramid

Drag Experiment

To imitate a more complex manipulation of the crust using the algorithms, a second experiment was conducted. In the first step of this experiment, a user pushed the crust up 10 mm. Next, the user dragged his or her finger across three adjacent unit cells of the crust causing a ridge in the surface. The dragging motion was performed in five steps: two 5 mm height displacements followed by three displacements for each of the adjacent unit cells equivalent to 10 mm [12]. The height or z-displacements of the unit cells along the bottom left-side edge and the bottom right-side edge were fixed. Also, the corner unit cells in Figure 56 were fixed in all directions. For this experiment, the tested crust sizes were increased by increments of 2-by-2. The smallest array of unit cells tested was 8-by-8 and the largest was 22-by-22.

The resulting crust shape for a 10-by-10 array of unit cells is shown in Figure 56. This shape appeared reasonable based on the similarities in configuration to the crust shape produced in the push experiment and because the links between the unit cells were slightly stretched instead of compressed. The green-dashed lines in Figure 56 are the initial positions and the solid-blue lines are the positions after the user pushed up on the crust.

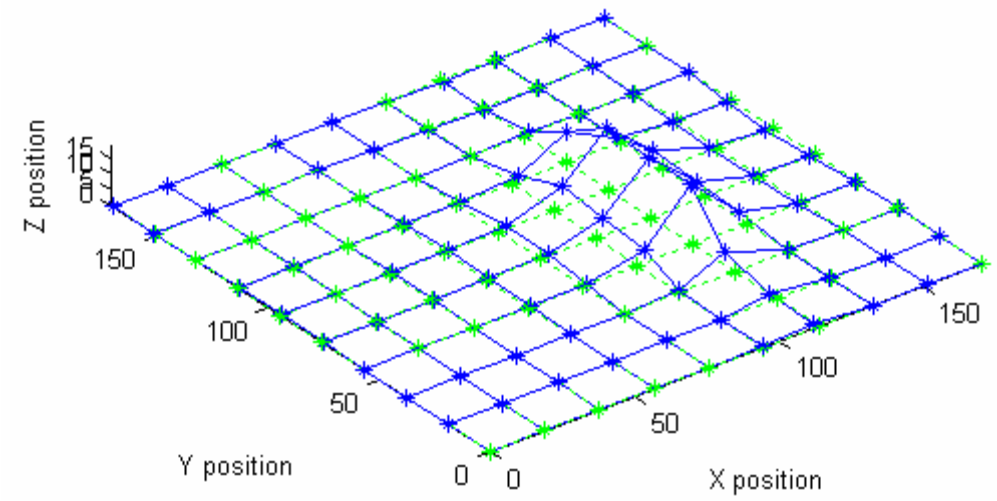


Figure 56 - Results from Algorithm 2 of the drag experiment on a 7-by-7 array

The average computation times of the drag experiment for the two algorithms are shown in Table 3. As expected, the total elapsed times for this experiment were too slow for real-time estimation with values nearly double the amounts of the push experiment. The average times per iteration are linear to the average times per iteration of the push experiment, which was anticipated. When the x-coordinates and the y-coordinates were constrained, the time was recorded in the last column of table. The beginning iterations took longer than the iterations without the constraints, but as the arrays increased in size the difference between the iteration times decreased.

For this experiment, the R^2 values for both Algorithm 1 and Algorithm 2 were very high with the third order being equal to one.

Table 3 – Average computation times for the two algorithms during the drag experiment

	Algorithm 1		Algorithm 2		X & Y Boundary Condition	
Crust Size	**Time per iteration (sec)**	**Elapsed Time (sec)**	**Time per iteration (sec)**	**Elapsed Time (sec)**	**Time per iteration (sec)**	**Elapsed Time (sec)**
8-by-8	0.102597	5.343265	0.121878	6.588511	0.137241	10.278111
10-by-10	0.213647	13.04159	0.245752	15.369855	0.270034	19.170478
12-by-12	0.384628	24.348255	0.532315	32.781014	0.552138	34.653570
14-by-14	0.695024	45.625495	0.900562	58.567465	0.922614	60.453899
16-by-16	1.213138	74.267507	1.50659	98.176656	1.527527	100.056151
18-by-18	1.904843	121.507588	2.374489	158.989712	2.395984	160.872676
20-by-20	3.002311	191.922682	3.777679	245.310847	3.798895	247.192076
22-by-22	4.316412	286.409732	5.541752	370.016235	5.563107	371.898332
R^2, 2nd order	0.9995	0.9992	0.998	0.9997	0.998	0.9997
R^2, 3rd order	1	1	1	1	1	1

Review of the Algorithm Tests

In this chapter two types of experiments were constructed to examine the disposition of the algorithms. The first test revealed that a symmetric solution to the mathematical model is dependent on two variables. The first revelation is by increasing the number of constraints, the solution goes toward a symmetric result. The second variable that influenced the capability of obtaining a symmetric solution was increasing the accuracy of the initial positions of the unit cells of the crust.

Now that the algorithms have been introduced, discussed, and tested, the mathematical models can be formulated to output a unique solution. For the mathematical models to portray the fabricated arrays, some physical properties needed to be implemented into the mathematical models. The next chapter discusses the process that was followed to acquire the spring constants of the leaf spring and of the spherical compliant joint.

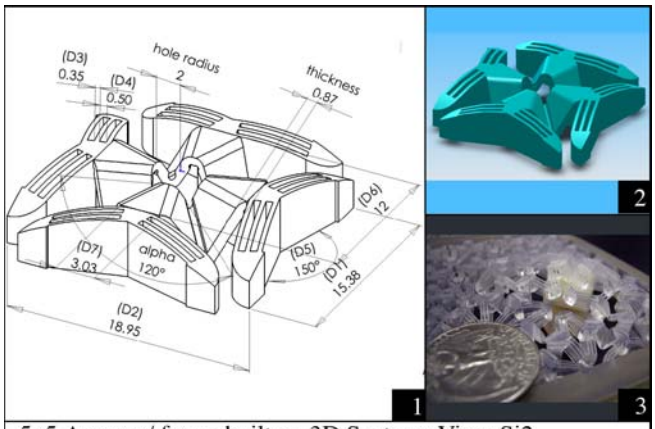
CHAPTER 7

CALCULATING THE SPRING CONSTANTS

A common assumption in conventional structural analysis is that joints are either perfectly rigid or perfectly hinged. Even though actual structures do not exhibit this behavior, the joints are represented as fictitiously fixed or hinged points. For a more accurate analysis of a structure it would be advantageous to include the true behavior of the joints. For this thesis, the structures being analyzed are the components of the unit cells: the leaf springs, and the spherical joints. The joints of the unit cells can be described as semi-rigid joints. Semi-rigid joints can be modeled using springs and the stiffness of the spring, k, can be determined by testing. In the case of rotational springs, the stiffness is characterized by the slope of a moment-rotation curve; whereas the translational springs stiffness constants are characterized by the slope of a force-distance curve.

For the mathematical models to accurately portray the physical properties of the unit cells, it was necessary to determine the spring constants. This chapter discusses how the values of the spring constants for the leaf springs and the spherical joints where obtained by creating a finite element analysis (FEA) model of the unit cells. The unit cell design used in this chapter, shown in Figure 57, was the design developed by Becker and discussed previously in Chapter 4. A force was applied to the FEA model using the ANSYS software package to determine what displacements occurred to the unit cells. To verify the data obtained from the FEA model, experimental values were obtained by testing several unit cells that were manufactured using stereolithography. Since resin material properties can vary, a range was used in the FEA modeling for Young's Modulus and Poisson's Ratio, but the force applied to the unit cells was kept constant.

The ranges used for Young's Modulus and Poisson's Ratio will be discussed in more detail the following sections.



5x5 Array w/ frame built on 3D Systems Viper Si2	
Mode: High Res	Room Temp: 70F
Line Width Compensation: Auto	Room Humidity: 58%
Z Correction: Auto	Vat Temp: 25.3C
Resin: SL-7545	Laser Power: 63mW

Figure 57 – Spherical compliant joint unit cell with flexible joints

Theoretical Spring Constant

The spring constant values of the unit cell were first investigated theoretically by exploiting an FEA model in ANSYS. FEA was first developed in 1943 by Richard Courant to obtain approximate solutions to vibration systems utilizing the Ritz method of numerical analysis and minimization of variational calculus [41]. Soon after this, a more comprehensive approach to element analysis was developed based on the concepts for stiffness and deflection of complex structures.

FEA consists of a computer model of a designed part or material that is analyzed for specific results. There are three basic types of analysis: 2-D modeling, Generalized Plane Deformation (GPD), and 3-D modeling [41]. 3-D modeling is the only analysis utilized in this thesis. It produces more accurate results but requires longer computational time than 2-D modeling and GPD.

ANSYS Modeling

The 3-D CAD model of the unit cell was imported into ANSYS as an IGES file. The IGES format serves as a neutral data format to transfer the design to a different system or program. The CAD model was imported into ANSYS in two phases. Because of the symmetric design of the unit cell, a fractional model for each phase was utilized. The first phase involved an eighth model representation of the unit cell that was employed to find the spring constant of the leaf springs. The second phase handled a quarter model representation of the unit cell to determine the spring constant of the spherical joints.

The most distinctive feature of the finite element method (FEM) that separates it from other methods is the division of a given domain into a set of simple subdomains, called elements. Both ANSYS models investigated in this thesis were configured using SOLID45 elements. As shown in Figure 58, SOLID45 elements are used for 3-D models of solid structures. The elements are defined by eight nodes having three degrees of freedom at each node that are represented as translations in the x, y, and z-directions [42].

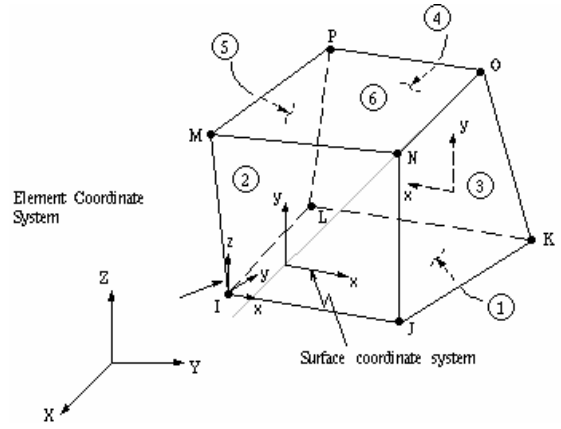


Figure 58 – SOLID45 3-D structural solid element

FEA utilizes a grid system called a mesh that is composed of the nodes that form the elements of the part. The mesh is designed to embody the material characteristics and

the structural properties which define how the model will react to loading conditions. This is achieved in ANSYS by defining the material model behavior. The material model behavior for both FEA models were defined as follows: structural, linear, elastic, and isotropic.

The next step involved identifying the values of Poisson's ratio and Young's Modulus for the resins. Two different resins were utilized to manufacture the crust: DSM Somos Waterclear 10120 and RenShape SL 5510. As mentioned earlier material properties of resins can vary, so a range of the values for Poisson's ratio and Young's Modulus were applied in the FEA modeling. The range of Poisson's ratio was kept constant for the two different resins. The first value of Poisson's ratio, 0.3, was obtained by utilizing the information provided on a material data sheet of an epoxy acrylic composite [43]. The second value, 0.4, was obtained from averaging a range obtained in a literature search [44]. The value range of Young's Modulus for WaterClear 10120 was 1709 MPa to 1960 MPa [45, 46]. Additionally, the value range utilized for SL 5510 was 2854 MPa to 3296 MPa [47, 48].

The next two sections describe how the two FEA models were constrained and loaded to determine the theoretical spring constants of the leaf spring and the spherical joint.

Leaf spring

The symmetry of the unit cell allowed for an eighth model depiction to be utilized in ANSYS. After the IGES file was imported and the material properties were entered into the FEA model, the part needed to be meshed. As stated previously, the SOLID45 element was applied to the model. A total of 13629 elements were constructed to create the mesh of the model.

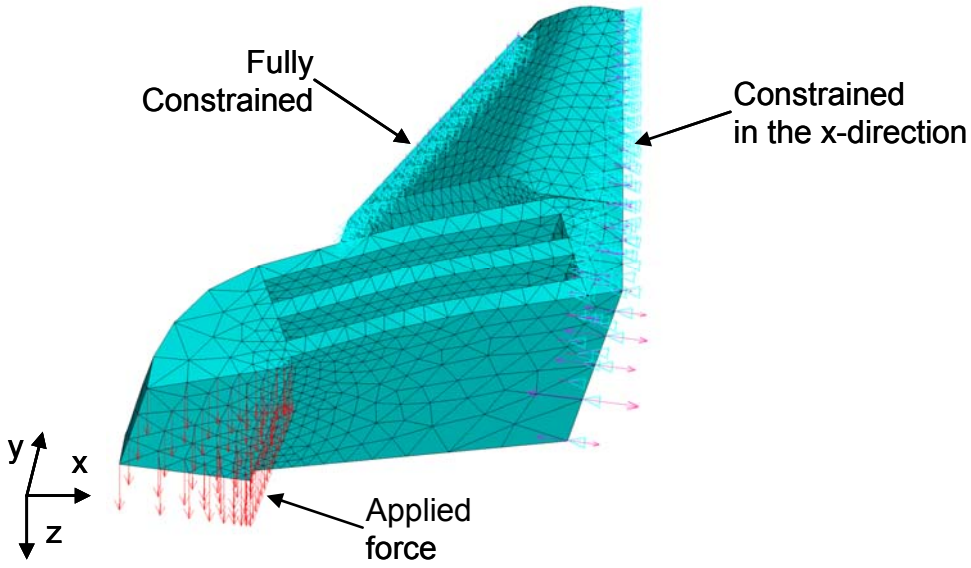


Figure 59 – An eighth model of the unit cell with boundaries

Two different boundary conditions and one force were applied to the FEA model as shown in Figure 59. The boundary conditions used in this model were set to describe the symmetry plane and to fix one plane of the structure to prevent free rotation. The fixed plane was also implemented to concentrate the displacement of the model to the leaf spring. As shown in Figure 59, the symmetry boundary condition restricts movement in the x-direction while the fixed boundary condition limits motion in all directions.

Before the loading conditions were added to the model, a displacement boundary condition in the z-direction was implemented on the model where the force was to be applied. After the model was evaluated in ANSYS, the total required surface force for the z-displacement boundary condition was obtained. The force was then divided by the number of nodes on the applied surface and re-applied on the surface as nodal forces. The intention of this procedure was to ensure the applied nodal force would create the same amount of movement as the displacement boundary condition.

The model was tested by applying different nodal forces in the z-direction to the surface of the leaf spring as illustrated in Figure 59. An example of three tested forces, all units are in Newtons, are shown in Table 4. The nodal forces were calculated by applying Newton's second law:

$$F_{nodes} = (ma)/nodes \qquad (23)$$

where F_{nodes} is the average force per node applied to the leaf spring surface, m is the amount of mass to be attached to the unit cells, a is equal to the acceleration due to gravity, and *nodes* is the total number of nodes that are contained in the surface.

Some of the numerical results of the FEA model utilizing SL 5510 as the material are shown in Table 4 as displacements. Also shown in the table are the model input values of Poisson's ratio (n), Young's Modulus, the total force, and the nodal force.

Table 4 – Data inputs for SL 5510 and calculated displacements for the FEA model

n	Young's (MPa)	Weight (g)	Force (N)	Nodal Force (N)	ANSYS displacement (mm)
0.4	2854	23.1	2.27E-01	2.80E-03	**0.0936**
		52.01529052	5.10E-01	6.30E-03	**0.211**
		186.0550459	1.83E+00	2.25E-02	**0.753**
0.3	2854	23.1	2.27E-01	2.80E-03	**0.1075**
		52.01529052	5.10E-01	6.30E-03	**0.227**
		186.0550459	1.83E+00	2.25E-02	**0.811**
0.4	3296	23.1	2.27E-01	2.80E-03	**0.0811**
		52.01529052	5.10E-01	6.30E-03	**0.1825**
		186.0550459	1.83E+00	2.25E-02	**0.652**
0.3	3296	23.1	2.27E-01	2.80E-03	**0.0873**
		52.01529052	5.10E-01	6.30E-03	**0.1965**
		186.0550459	1.83E+00	2.25E-02	**0.702**

The graphical results shown in Figure 60 were obtained by plotting the nodal solutions in ANSYS. It can be seen from the figure that the upper-portion of the unit cell, which is part of the spherical joint, does not displace. This indicates that all of the displacement occurred in the leaf spring portion of the unit cell.

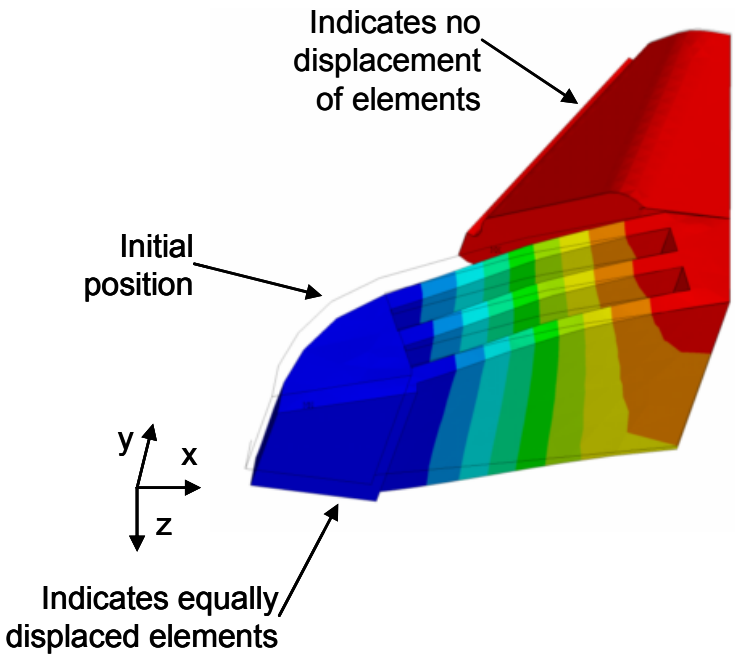


Figure 60 – Illustration of the resulting displacement of the unit cell in ANSYS

The output information obtained from the analysis of the FEA model was the displacement of the unit cell. The spring constant ranges were calculated by dividing the forces applied by the displacements obtained from the FEA model. The calculated spring constant ranges and the displacement of the unit cell are shown in Table 5.

Table 5 – Displacement and spring constant value ranges obtain from the FEA model

LEAF SPRING VALUES			
WaterClear 10120		SL 5510	
Range of displacement (mm)	Spring Constant Range (N/m)	Range of displacement (mm)	Spring Constant Range (N/m)
0.167 - 0.222 0.37 - 0.5 1.32 - 1.785	1020.54 - 1382.727	0.22139 - 0.27515 0.44324 - 0.55087 0.67901 - 0.84389	1868.079 - 2321.893

Spherical Joint

The eighth model depiction utilized during the determination of the spring constant range for the leaf spring was not applied during the investigation of the spherical

joints. Instead, a quarter model depiction was applied to incorporate the coupled nature of the opposing spherical joints and to better duplicate the force the spherical joints will experience. As with the eighth model, an IGES file was imported and the same element type was applied. No changes to the material properties were made because the same two resins were investigated. Since the quarter FEA model is larger, a larger amount of elements, 26805 elements, were constructed to create the mesh model.

Similar to the eighth model, two different boundary conditions and one force were applied to the quarter FEA model as shown in Figure 61. The boundary condition induced because of symmetry restricts movement in the z-direction and the fixed boundary condition limits displacement in all directions.

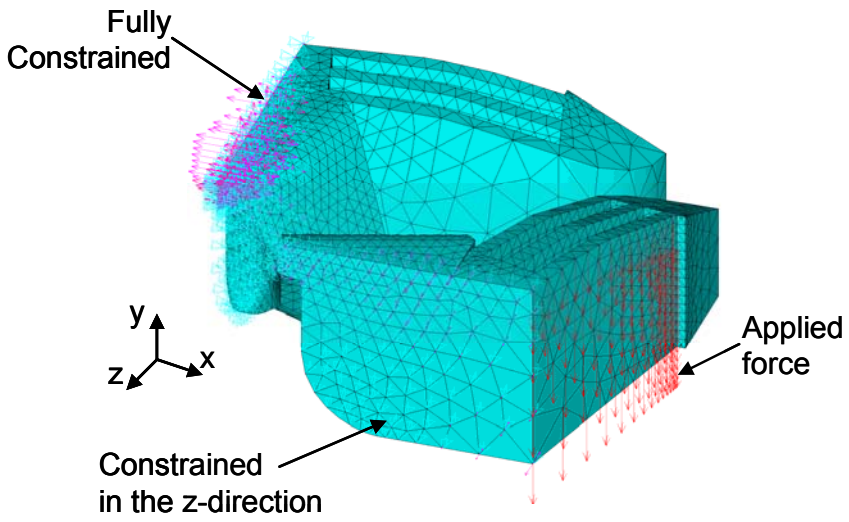


Figure 61 – An eighth model of the unit cell with boundaries

As with the previous FEA model, a displacement boundary condition was implemented in place of the force to test the model. After testing the movement of the displacement boundary against the displacement incurred from the applied nodal forces, different forces were administered to the model. The results obtained from the analysis of the model were in the form of displacements.

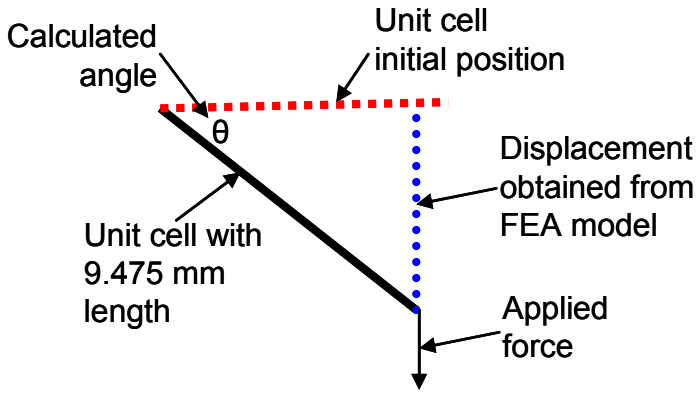


Figure 62 – Illustrates triangle side values used to calculate the change in angle

The changes in angles created by the applied forces were calculated by utilizing the Pythagorean theorem. As illustrated in Figure 62, two sides of a right triangle are known. The first side, shown as a solid black line, is the distance from the center of the unit cell to the end of the leaf spring, which equals 9.475 mm. The value of the second side, illustrated as a dotted blue line in Figure 62, is the displacement obtained from the FEA model. Some of the numerical results of the FEA model utilizing SL 5510 as the material are shown in Table 6.

Table 6 - Data inputs, displacements, and angles for the FEA model utilizing SL 5510 as the material

n	Young's (MPa)	Weight (g)	Total Force (N)	Nodal Force (N)	ANSYS Displacement	Angle (radians)	Angle (degrees)
0.4	3296	108.3	1.062	1.34E-02	1.251	0.11455	6.56330
		212.4	2.084	2.64E-02	2.438	0.22457	12.86715
		317.3	3.113	3.94E-02	3.636	0.33868	19.40496
		369.7	3.627	4.59E-02	4.234	0.39728	22.76255
		422.1	4.141	5.24E-02	4.832	0.45736	26.20493
		474.5	4.141	5.24E-02	4.832	0.45736	26.20493
0.3	2854	108.3	1.062	1.34E-02	1.337	0.14158	8.11198
		212.4	2.084	2.64E-02	2.607	0.27874	15.97066
		317.3	3.113	3.94E-02	3.886	0.42260	24.21312
		369.7	3.627	4.59E-02	4.526	0.49801	28.53386
		422.1	4.141	5.24E-02	5.165	0.57653	33.03278
		474.5	4.141	5.24E-02	5.165	0.57653	33.03278

A graphical result of the unit cell with an applied force is shown in Figure 63.

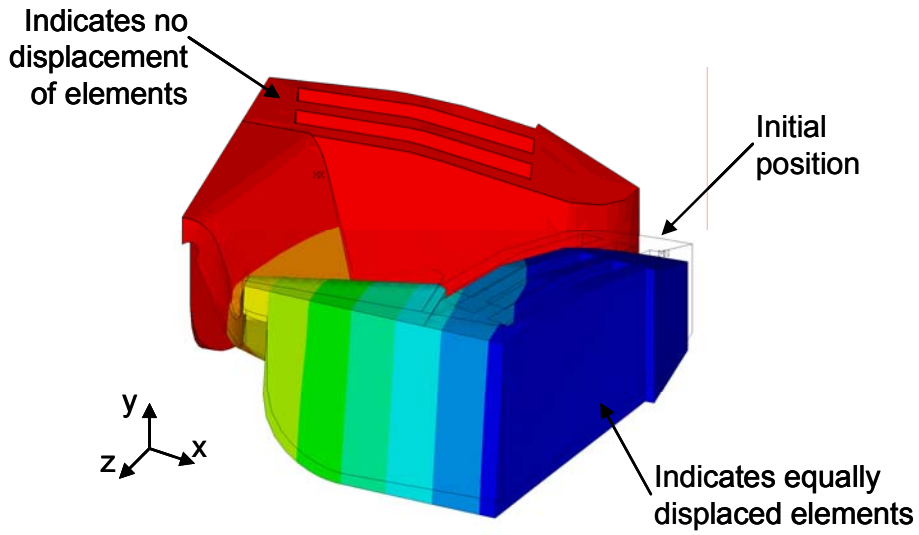


Figure 63 - Results illustrating the displacement of the unit cell in ANSYS

To calculate the average spring constant of the spherical joints, the torque needed to be calculated because of the bending moment being introduced to the unit cell. The values of torque were calculated by multiplying the applied total force by the distance 9.475 mm. Next, the computed torque values were divided by the change in angle. The resulting calculated spring constant ranges are shown in Table 7.

Table 7 – Displacement and spring constant value ranges obtain from the FEA model

SPHERICAL JOINT VALUES			
WaterClear 10120		SL 5510	
Change in Angle (degrees)	Spring Constant Range (Nm)	Change in Angle (degrees)	Spring Constant Range (Nm)
10.293 - 13.314 21.555 - 26.556 33.423 - 42.089	0.0397 - 0.0527	6.563 - 8.112 12.867 - 15.971 19.405 - 24.213	0.0681 - 0.0879

Experimental Spring Constant

To verify the data obtained from the FEA models, experimental spring constant values were obtained by testing several unit cells that were manufactured using

stereolithography. To find the spring constant of the manufactured surface, single unit cells and 2-by-1 arrays of the unit cells were fabricated. See Figure 64 for a display of

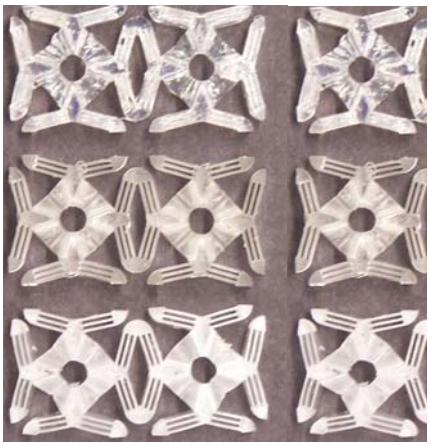


Figure 64 – Fabricated 2-by-1 arrays (left) and single unit cells (right) used for spring constant experiments

the fabricated unit cells. Two different resins were used during the fabrication process: WaterClear 10120 and SL 5510. The manufacturing process and post processing for each build was conducted in the same manner for all fabrications discussed in this thesis. After the unit cells were constructed in a 3D Systems Viper Si2 machine, they were post-processed for ten minutes in a Branson 5210 ultrasonic cleaner while soaking in isopropyl alcohol. This was followed by fifteen minutes in an ultraviolet oven to further cure the resin. Before the fabricated unit cells were subjected to testing, a seven day waiting period was allocated. This helped to ensure that the degradation of the resin material would be minimal. Additionally, the fabricated unit cells were stored in a dark dry area to prevent ultraviolet light from influencing the mechanical properties of the unit cells. The arrays of unit cells were then subjected to experimentations that imitated the loading of the FEA model analysis to obtain the spring constants of the leaf springs and the spherical joints of the unit cells. This section will describe how the "hang test" procedure was used to determine the spring constants of the unit cells.

Determining the Spring Constant of a Hanging Spring

Hooke's law was utilized to experimentally ascertain the spring constants of the unit cells. Hooke's law states that the stress of a solid is directly proportional to the strain applied to it. In this case, Hooke's law gives the relationship between the force applied (stress) to an unstretched spring and the amount the spring is stretched (strain) when the force is applied. Hooke's law is:

$$\boldsymbol{F} = -k\boldsymbol{x} \qquad (24)$$

where $\boldsymbol{F}$ equals the force applied, k is the spring constant, and $\boldsymbol{x}$ is the displacement. Rearranging equation *(24)* and investigating just the magnitude of the expressions gives:

$$k = F/x \qquad (25)$$

The spring constant can be calculated by applying a known force to a spring and measuring the displacement of the spring derived by the applied force. One way to calculate the spring constant of a spring is by completing hang tests on the spring. The hang test procedure involves suspending a spring from a support and loading weight to the spring. After the weight is loaded, the displacement of the spring is measured and recorded. This procedure is illustrated in Figure 65. Typically, as more mass is introduced to the spring, more displacement will occur. Once a collection of data points is acquired, a visual trend of the behavior of the acquired data points can be produced.

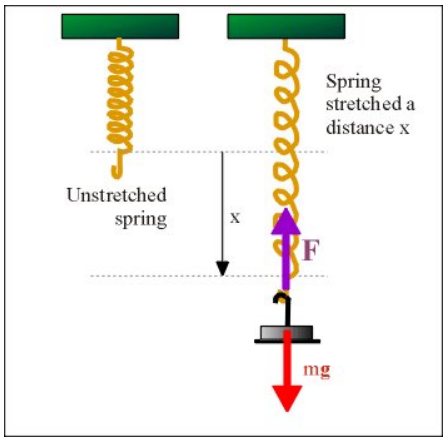


Figure 65 - A hang test experiment to calculate the spring constant

A linear regression or best-fit of the acquired data points is completed utilizing the method of least-squares. By observing the experimental data and the best-fit line, it can be verified whether or not Hooke's Law is applicable. The coefficient of determination, R^2, describes quantitatively how well the best-fit line represents the obtained data points. Hooke's Law is applicable if the best-fit line represents the data points. If it is applicable, it signifies the force applied is directly proportional to the displacement of the unit cell and equation *(25)* can be utilized to obtain *k*.

The slope of the linear regression line obtained from a force versus displacement graph corresponds to the spring constant, *k*. The equation of a straight line in slope-intercept form is as follows:

$$y = mx + b \tag{26}$$

where *y* is the ordinate, m is the slope, *x* is the abscissa, and b is the *y*-intercept of the line. For a spring still within its elastic limit, Hooke's Law gives the relationship between the force, *F*, and the spring displacement, *x*, as:

$$F = kx \tag{27}$$

A direct comparison between equation (26) and equation (27) shows the following:

$$y \rightarrow F \tag{28}$$

$$x \rightarrow x \tag{29}$$

$$m \rightarrow k \tag{30}$$

$$c \rightarrow 0 \tag{31}$$

By observation, it can be understood that the slope of the linear regression line corresponds to the spring constant of the spring.

Leaf Spring

As with the FEA model, the leaf spring was the first component of the unit cell to be investigated utilizing fabricated 2-by-1 arrays of the unit cells. Similar to the hang test procedure previously described, the unit cells were secured to a backing. The backing

was placed on a surface and a level was used to draw horizontal lines on the support for placement of the fabricated arrays. The 2-by-1 array was adhered vertically to the backing by aligning the triangular points, outlined in black in Figure 66, of the unit cell with the measured horizontal line.

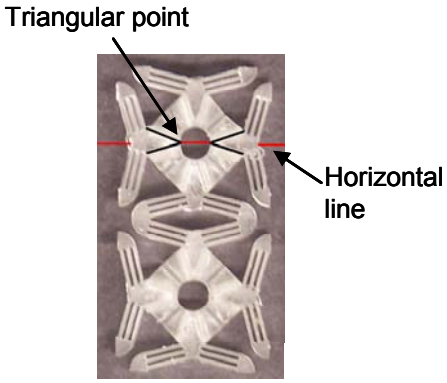


Figure 66 – Illustration of the triangular points utilized to align the unit cell on the support

A hanger device was created to attach weights to the arrays. The placement of the hanger was critical because it was desired to only engage the leaf spring to displace. This was achieved by attaching the hanger underneath the leaf spring of the unit cell. A series of weights were measured on a balance and recorded in a data base. They were then placed on the hanger device causing the leaf spring to stretch. Before the spring displacements were measured, a ten minute waiting period was allotted to allow the system to reach an equilibrium state. Each measurement that was made has some sort of unavoidable error attached to it. Whether the errors were random or systematic, only relying on one measurement means that the results can be highly uncertain. To ensure accurate measurements of the displacements, two different regions of the unit cells were marked and measured using a set of digital calipers. From the measured displacements, an average displacement was calculated and recorded for each weight that was added to

the unit cell. Figure 67 illustrates the configuration of the array of unit cells with the hanger and regions marked for displacement measurements.

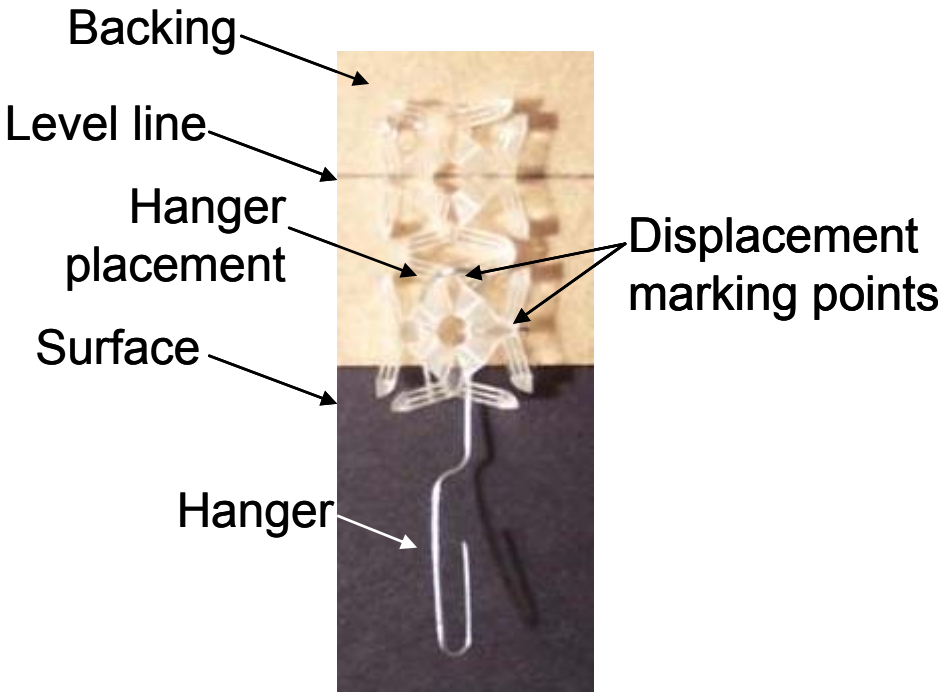


Figure 67 – The 2-by-1 array test layout for finding the spring constant for the leaf spring

Different amounts of masses were hung on nine separate arrays of unit cells to obtain several contrasting values of displacement corresponding to the various values of the applied forces. Each time a new force was added to an array of unit cells, a ten minute waiting period was allocated for the system to reach equilibrium before measurements were taken. For the leaf spring portion of a unit cell fabricated using SL 5510, the value of the spring constant was obtained using the slope of the regression line shown in Figure 68. The vale of the slope was 1933 N/m. When performing any type of best fit analysis, it is imperative to know quantitatively how appropriate the regression line is compared to the acquired data. The R^2 value, known as the coefficient of determination, shown in Figure 68 describes quantitatively how well the regression line fits the data. In the analysis of the data shown in the figure, the R^2 value was calculated

to be 0.9986 which indicates the regression line is sensible. The obtained data set in its entirety is located in Appendix B.

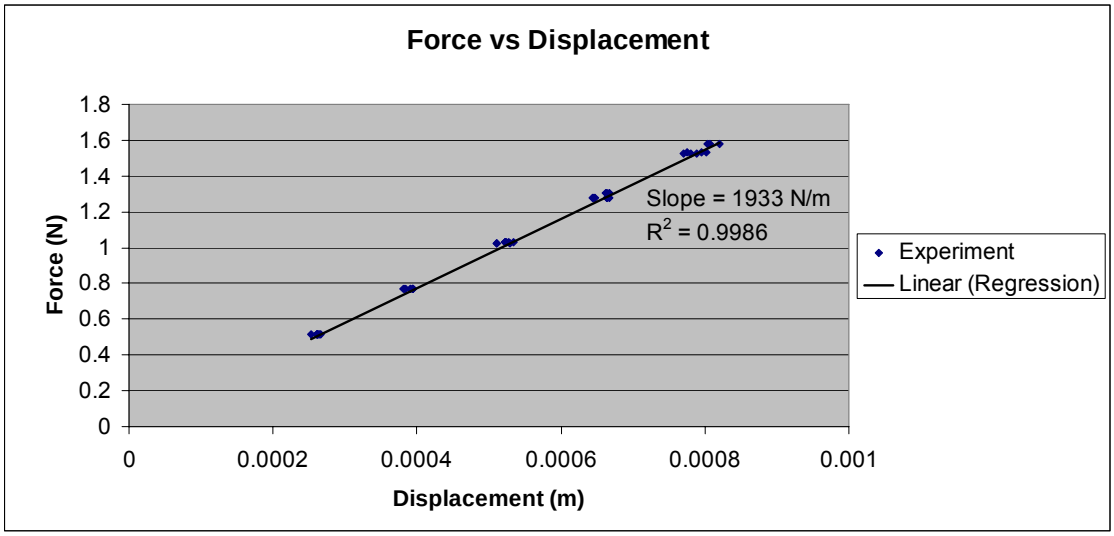


Figure 68 – Plot of applied force versus spring displacement for the leaf springs using SL 5510

The linear regression line shown in Figure 69 illustrates the stiffness value of a leaf spring fabricated using WaterClear 10120. The calculated value equals 1211 N/m. A visual inspection of the linear regression line compared to the experimental data indicates a sensible fit. This visual inspection was confirmed by the high R^2 value of 0.9976. The closer the R^2 value is to one, the better the regression line fits the data.

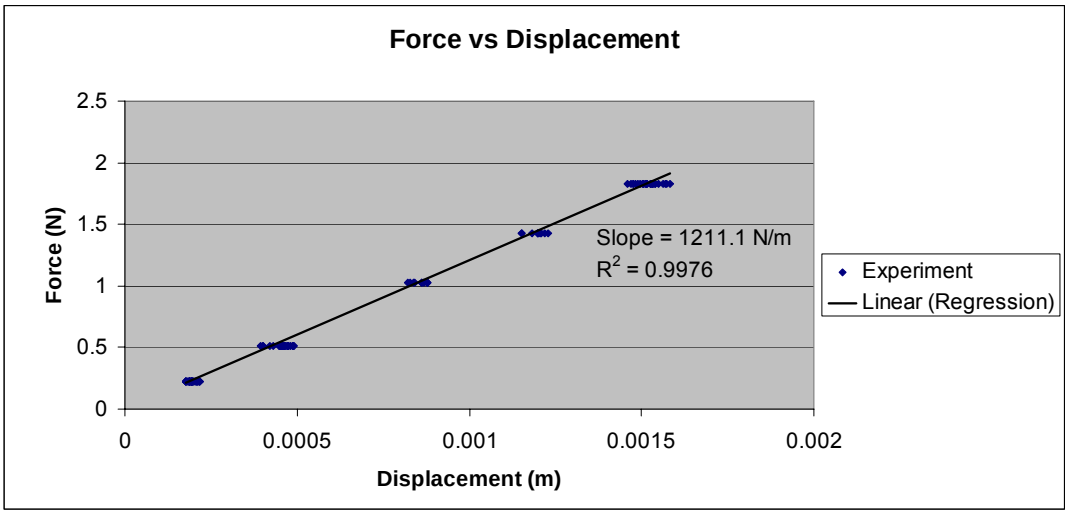


Figure 69 - Plot of applied force versus spring displacement for the leaf springs using WaterClear 10120

Spherical Joint

The spherical joint was tested in a manner similar to the leaf spring tests. A level was used to create even lines on the backing to ensure the single unit cells were adhered accurately to the mounting surface. The backing was then mounted to a surface and weights were hung from the unit cells. Figure 70 depicts the experimental layout for testing the spherical joints.

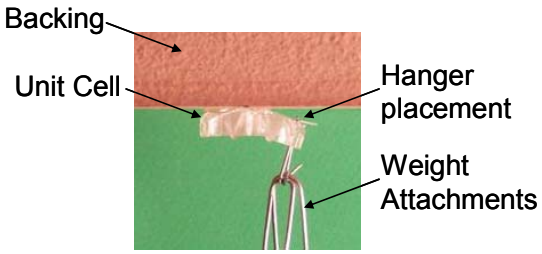


Figure 70 - The unit cell test layout for finding the spring constant for the spherical joints

Obtaining an accurate measurement of the angle produced by applying a force on the end of the unit cell proved to be difficult. Since the angle could not be accurately measured by using a protractor because of the small nature of the unit cell, a program was

developed to analyze the angle between selected objects in a picture. A digital camera was setup on a tripod and adjusted to the same height as the support and unit cell combination. Next, a level was used to ensure the camera was capturing a straight picture of the unit cell apparatus. After the pictures were taken, they were uploaded into the analysis program to calculate the angular displacement of the unit cells.

As mentioned in previous chapters, the spherical joints were represented mathematically as rotational springs. A rotational spring is characterized by an applied torque versus angular rotation curve. The spring constant, k, is equal to the slope of the curve. Since the angular displacement was calculated by analysis of digital pictures, the next data needed to determine the value of the spring constant was the applied rotational forces that corresponded to the angular rotation. The applied force multiplied by the distance of the applied force to the fulcrum equals the value of torque.

Figure 71 illustrates a linear regression line with a slope equal to 0.0775 Nm. The value of the slope was calculated from the collected data points involving the displacement angles of unit cells fabricated from SL 5510 and the corresponding torque values. The spring constant of a spherical joint fabricated from SL 5510 resin equals 0.0775 Nm, which is the value of the slope of the linear regression line. The appropriateness of the linear regression line to the acquired data was confirmed by the relatively high R^2 value of 0.9958.

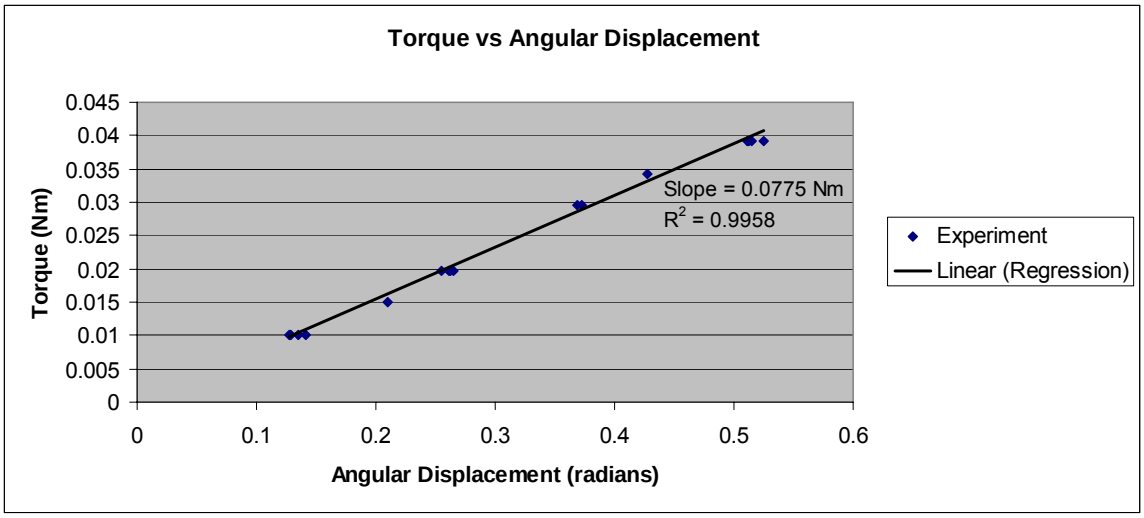


Figure 71 – Plot of torque versus angular displacement of the unit cell fabricated using SL 5510 resin

The graph shown in Figure 72 was produced from data points acquired from testing a spherical joint fabricated from WaterClear 10120. The slope of the linear regression line, which is also equivalent to the spring constant of a spherical joint, was determined to equal 0.0518 N/m. The value of the coefficient of determination signifies that the linear regression line is an acceptable fit to the data.

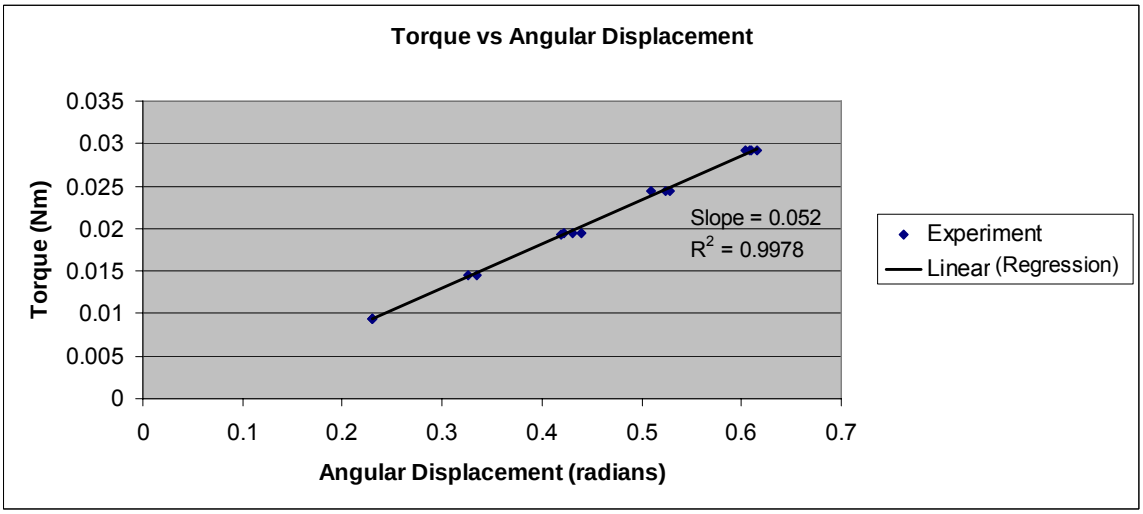


Figure 72 - Plot of torque versus angular displacement of a unit cell fabricated using WaterClear 10120

Data Comparison

The calculated spring constants and the experimentally measured displacements of the leaf spring are comparable to the values obtained analyzing the FEA model because all of the measured spring constants are located within the range of the ANSYS results.

Table 8 shows a comparison between the experimentally obtained data and the FEA model data for the leaf spring constants. The measured experimental spring constant of the leaf spring was located approximately in the middle of the range obtained using ANSYS. This implies a strong correlation between the measured data and the theoretical data. For the spring constant of the leaf spring fabricated using SL 5510, the experimental value is offset from the center of the data range located toward the lower end of the ANSYS results.

Table 8 – Comparison of the outputs obtained theoretically and experimentally for the leaf spring

	LEAF SPRING VALUES			
	WaterClear 10120		SL 5510	
	FEA Model	Experimental Data	FEA Model	Experimental Data
Range of displacement (mm)	0.167 - 0.222 0.37 - 0.5 1.32 - 1.785	0.1765 - 0.216 0.4 - 0.489 1.46 - 1.581	0.22139 - 0.27515 0.44324 - 0.55087 0.67901 - 0.84389	0.2535 - 0.266 0.511 - 0.534 0.7705 - 0.802
Spring Constant Range (N/m)	1020.54 - 1382.727	1211.1	1868.079 - 2321.893	1933

As with the data comparison of the leaf spring values, the theoretical values of the spring constant and angular displacement of the unit cell are comparable to the values obtained experimentally, as shown in Table 9. Both measured values of the spring constant are located within the range obtained using an FEA model. The measured experimental spring constant for the unit cell fabricated using WaterClear 10120 is located on the higher end of the ANSYS results. Using SL 5510 during the

manufacturing process created the documented experimental spring constant to be equal to 0.0775 N·m, which is located in the middle of the ANSYS results.

Table 9 – Comparison of the outputs obtained theoretically and experimentally for the spherical joint

	SPHERICAL JOINT VALUES			
	WaterClear 10120		SL 5510	
	FEA Model	Experimental Data	FEA Model	Experimental Data
Change in Angle (degrees)	10.293 - 13.314 21.555 - 26.556 33.423 - 42.089	13.18 - 13.2 23.995 - 25.2 34.62 - 35.25	6.563 - 8.112 12.867 - 15.971 19.405 - 24.213	7.27 - 8.09 14.6 - 15.15 21.13 - 21.35
Spring Constant Range (Nm)	0.0397 - 0.0527	0.0518	0.0681 - 0.0879	0.0775

As shown in the two tables, the theoretical data was verified by the data that was obtained experimentally. The acquired experimental spring constant was inputted into the mathematical model to more accurately portray the physical traits of the unit cells. A comparison of the mathematical models and fabricated crust will be investigated in the next chapter.

Review of the Spring Constants

In this chapter the values of the spring constants were determined through two different methods: measured experimental data, and theoretical analysis utilizing a FEA model. Since the physical properties of the material used in the fabrication process vary, the value of Poisson's ratio and the value of Young's Modulus for the FEA model was given in a range to determine what the spring constant values were. The values utilized in the range were acquired from various sources of literature [43-48]. Additionally, two different materials were utilized to fabricate the unit cells needed for determining the measured experimental data. These two materials were WaterClear 10120 and SL 5510. All of the spring constant values obtained experimentally were located within the spring

constant value ranges calculated from the FEA model using ANSYS. Table 10 shows a summary of the spring constant values measured experimentally.

Table 10 – Summary of the spring constant values for the two components of the unit cell

Spring Constant Values			
WaterClear 10120		SL 5510	
Leaf Spring (N/m)	Spherical Joint (N/m)	Leaf Spring (Nm)	Spherical Joint (Nm)
1211.1	0.0518	1933	0.0775

The determined spring constant values were implemented into the mathematical model to improve the ability of it to mimic the physical properties of the fabricated surface. In the next chapter, a discussion comparing the mathematical model to the physical model is completed. The comparison was completed by testing the mathematical model to determine the z-coordinates of the unit cells and testing the fabricated array of unit cells under the same conditions as the mathematical model to discover the z-coordinates of the manufactured unit cells. The resulting z-values were then compared to determine the accuracy of the mathematical model.

CHAPTER 8

TESTING A MANUFACTURED MODEL

The experiments discussed in this chapter were performed to verify the predictive capability of the mathematical models to forecast the movement of a fabricated crust. The first comparison test was used to determine if a good correlation existed between unit cell position measured from a constructed surface and unit cell position calculated from the mathematical model. This test was achieved by constraining the unit cells in the x-direction and y-direction while displacing the centermost unit cell in the z-direction for the mathematical model and the fabricated surface. The second test was an examination into the behavior of the unit cells designed by Becker. The size of the surface was decreased to a 1-by-9 array of unit cells to determine if the leaf springs were obstructing the bending ability of the spherical joint. The procedure that followed was completed in the same manner as the first comparison test with the except that the fabricated surfaces were constructed from several different unit cell designs. The third test discussed in this chapter followed the same procedures used in the first comparison test with the exception of fewer constraints induced on the system.

First Comparison Test

The general procedure of the first experiment was very similar to the process followed in the push experiment that was discussed in Chapter 6. As with the push experiment, the four corner unit cells were fixed in all directions to prevent the crust from floating when a unit cell was actuated in the z-direction. The constraints induced on the system are shown in Figure 73. The fixed corners also helped the fabricated crust to form a distinct shape.

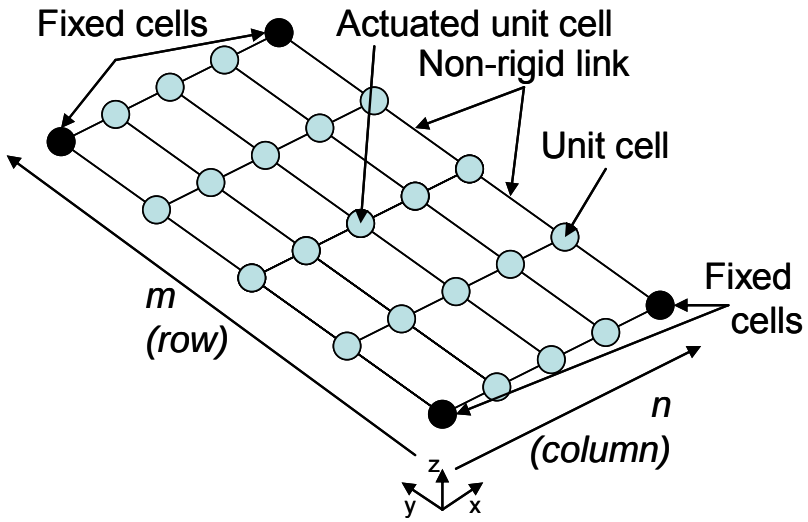


Figure 73 – Schematic of a 5-by-5 crust with constrained corner unit cells

An additional process step of the push experiment that was utilized in this first experiment involved actuating the unit cell located in the center of the crust several times to achieve different height displacements. There were three major differences between this comparison test and the push experiment. The first difference involved utilizing the calculated spring constants, discussed in the previous chapter, in the mathematical models. The second alteration to the push test was what output data was investigated. In the first comparison test, the actual unit cell positions were investigated instead of determining the amount of time the mathematical models took to iterate the positions of the unit cells. The last distinction between the two tests was the use of constraints. The x-coordinates and the y-coordinates were fully constrained in the first comparison test to enabled the mathematical model to produce a unique output solution for the z-coordinates of the surface.

Manufactured Surfaces

Two different sized crusts were fabricated from SL 5510 resin using a SLA Viper Si2 machine. An additional structure was manufactured around each crust to produce fixed constraints that were inflicted on the four corner unit cells of the surfaces. The

purpose of the constraints was to prevent the crust from free floating in space and to help produce a definitive shape from the crust that could be easily recorded. Each crust was manufactured and post-processed in the same manner as the fabricated unit cells discussed in Chapter 7.

The only difference between the two manufactured surfaces was the fabrication sizes; one crust was a 5-by-5 surface while the other was a 9-by-9 surface. The two fabricated surfaces are shown in Figure 74. Each surface array utilized the unit cell designed by Becker [37].

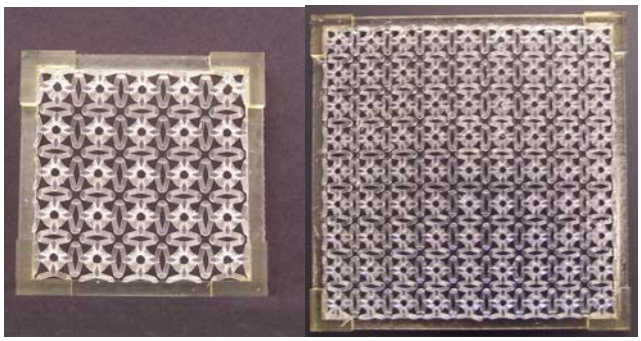


Figure 74 – Top view of a 5-by-5 (left) and 9-by-9 (right) crust surrounded by boundary conditions

To test the manufactured surface, a testing apparatus was constructed to secure the surfaces. The testing structure consisted of 25 straight rods attached to a base plate. The 25 rods were arranged to create a 5-by-5 array, as shown in Figure 75.

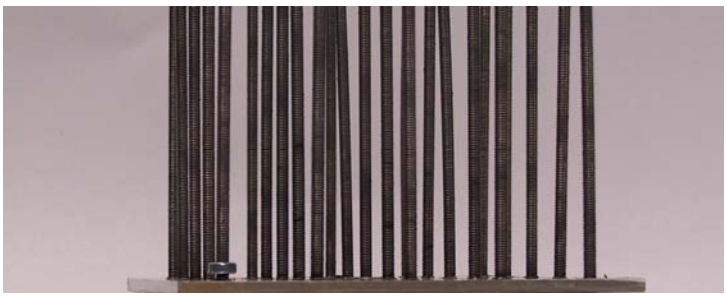


Figure 75 – Testing apparatus used to measure the displacement of the unit cells in the z-direction while inducing constraints in the x-direction and y-direction

The distance between each rod was 18.95 millimeters, which was equivalent to the distance between each unit cell. Each rod was threaded through the center of the 5-by-5 surface and the border surrounding the surface fit tightly on the base plate to prevent the corner unit cells from displacing. The surface was pushed onto the base plate until the bottom of the unit cells contacted the top of the base plate. The rods had three purposes. The first application of the rods was to constrain the unit cells in the x-direction and y-direction but still allow them to move in the z-direction. The next function of the rods was to displace one of the unit cells in a manner that could mimic the static displacement of an actuator. For the 5-by-5 surface, this was achieved by incorporating a threaded rod in the middle of the testing apparatus. A nut with a washer placed on top of it was utilized to displace the middle unit cell. The washer was sized such that it did not hinder the bending of the central unit cell. The nut was moved up and down the threaded rod to change the height of the displaced middle unit cell. Lastly, the rods were employed to measure the z-displacement of the unit cells. Once the centermost unit cell was displaced, the movement of the other unit cells was recorded by placing marks on the rods to represent the displacement. The markings were then measured using a caliper with the capability of measuring to the thousandths place. Figure 76 illustrates a 5-by-5 surface being displaced inside the testing apparatus.

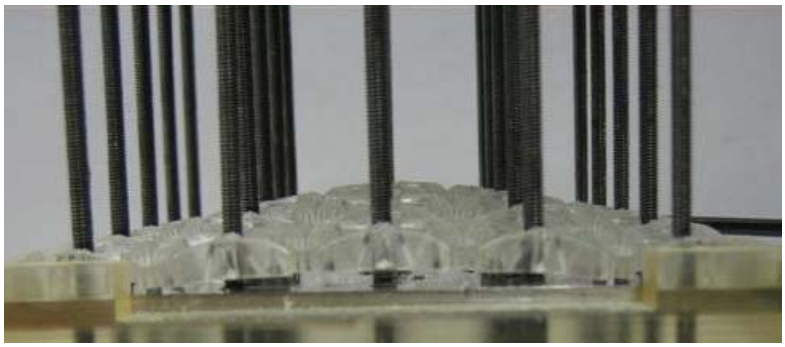


Figure 76 – A 5-by-5 surface constrained in the x-direction and y-direction and displaced in the center by 10 mm in the testing apparatus

The same procedure was followed for testing the 9-by-9 surface with a few essential differences. Only part of the 9-by-9 could fit on the testing apparatus, so not all of the unit cell z-coordinates could be measured. Due to the symmetric nature of displacing the centermost unit cell, this was not a problem. The threaded rod used to displace the central unit cell was relocated to one of the corner spaces. From this location, a quarter of the unit cell z-coordinates were able to be measured and the symmetric nature of the movement was taken into account. Besides not being able to measure all of the z-coordinates of the unit cells, the border surrounding the fabricated surface did not fit tightly around the base plate. Weights were placed on the corner points of the border to prevent the corners from lifting. The testing apparatus with a displaced 9-by-9 surface is shown in Figure 77.

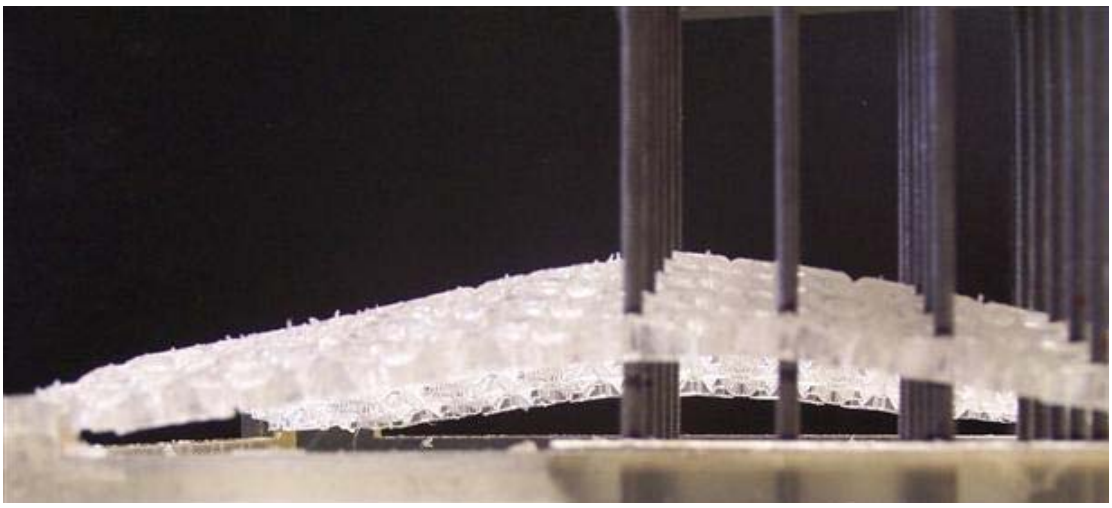


Figure 77 – A 9-by-9 surface with a quarter of the surface constrained in the x-direction and y-direction and displaced in the center by 20 mm in the testing apparatus

A comparison between the tested manufactured surfaces and the mathematical models needed to be completed to determine if the mathematical models accurately portrayed the movement of the fabricated surface. To complete the comparison, an equivalent test procedure needed to be accomplished for the mathematical models. The

corresponding test procedure for the mathematical models is discussed in the next section.

Mathematical Models

The x, y, and z-coordinates of the corner unit cells in the mathematical models were fixed to replicate the constraints inflicted on the manufactured crust by the surrounding boarder. Figure 78 illustrates the four constrained unit cells of a 5-by-5 crust and which unit cell was actuated in the z-direction. As with the manufactured surface, the actuated center unit cell was constrained in the x-direction and y-direction to prevent it from drifting in space. Furthermore, all x-coordinates and y-coordinates of the unit cells in the mathematical model were constrained to keep in compliance with the test conducted on the fabricated surface. After testing both mathematical models, it was determined that only one mathematical model was needed for this test. The reason for this is because the induced constraints did not allow the additional rotational springs in the second mathematical model, which were discussed in Chapter 5, to make a difference.

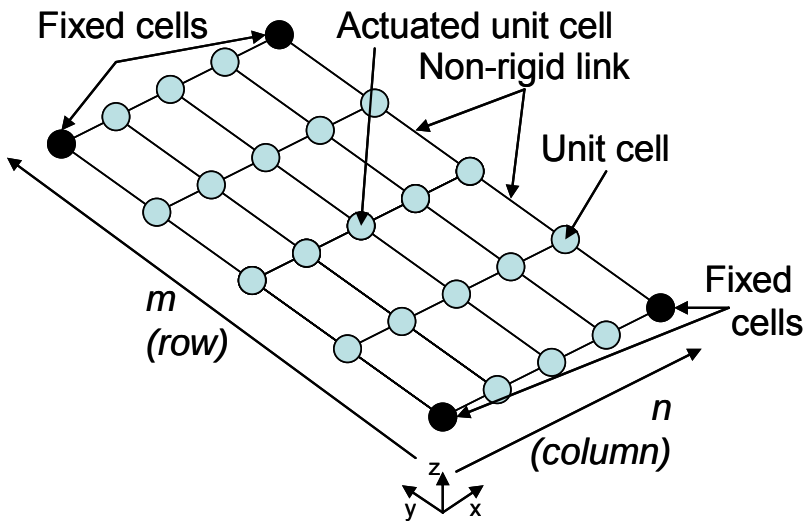


Figure 78 – Schematic of a 5-by-5 crust with constrained corner unit cells

In the next section, the acquired data from testing the manufactured surface and the mathematical models are introduced and compared.

First Comparison Test Results

A total of three fabricated surfaces was tested. The sizes of two of the constructed surfaces were 5-by-5 arrays. One of the 5-by-5 surfaces was fabricated using SL 5510 while the other 5-by-5 surface was manufactured from WaterClear 10120. The last fabricated surface was a 9-by-9 array composed from SL 5510. The obtained spring constants discussed in Chapter 7 were applied in the mathematical model.

The first surface that was tested was the 5-by-5 crust fabricated using SL 5510. The center unit cell of this surface was displaced 10 millimeters and 20.358 millimeters in the mathematical model and physically on the fabricated surface. The intention was to displace the surface a third time to 30 millimeters but the manufactured model broke due to the stiffness of the material used. Figure 79 illustrates the corresponding unit cells of Table 11. The acquired data for displacing the center unit cell by 10 millimeters is shown in Table 11.

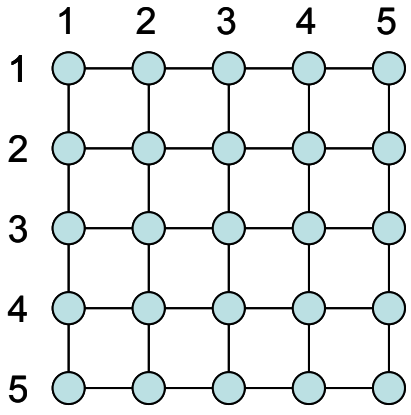


Figure 79 – Illustration of the corresponding unit cells of Table 11

Table 11 – The z-coordinates of a fabricated 5-by-5 surface displaced in the center by 10.0 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	2.319	3.357	2.319	0
2	2.314	6.335	7.835	6.35	2.319
3	3.374	7.906	10	7.929	3.39
4	2.306	6.308	7.882	6.355	2.298
5	0	2.308	3.366	2.308	0

	1	2	3	4	5
1	0	2.9999	4.4674	3	0
2	2.9999	5.0772	6.7986	5.0772	2.9999
3	4.4674	6.7986	10	6.7986	4.4674
4	3	5.0772	6.7986	5.0772	3
5	0	2.9999	4.4674	3	0

The acquired data for displacing the center unit cell by 20.358 millimeters can be seen in Appendix C. As can be seen in Table 11, both sets of data appear symmetric. A possible factor for the slight discrepancies between symmetric unit cells of the data set affiliated with the fabricated surface can be associated with measurement error or mark error. Even so, the two data sets do not correlate with each other. Some of the z-coordinates obtained from the displaced fabricated surface are lower than the z-coordinates calculated from the mathematical model while some of the z-coordinates are higher.

To ensure surface size was not responsible for the discrepancies between the fabricated surface and the mathematical model, the next surface investigated was the 9-by-9 crust fabricated from SL 5510. The middle unit cell of the 9-by-9 surface was displaced three times in the z-direction; 8.511 millimeters, 17.96 millimeters, and 26.215 millimeters. Figure 80 illustrates the layout of the unit cells of Table 12. The green-dashed circles represent the unit cells that were placed in the testing apparatus. The acquired data for displacing the center unit cell by 26.215 millimeters is shown in Table 12. The additional displacements can found in Appendix C.

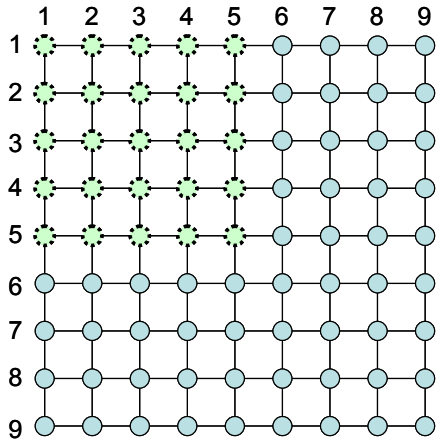


Figure 80 - The green-dashed circles indicate the unit cell placements that were measured from a 9-by-9 surface and recorded in Table 12

Table 12 – A quarter of the z-coordinates of a fabricated 9-by-9 surface displaced in the center by 26.215 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	4.818	8.138	10.541	11.125
2	4.754	8.28	11.638	14.902	15.664
3	8.453	11.808	17.078	19.092	19.705
4	10.51	15.125	19.364	23.141	23.695
5	11.188	15.6	19.908	23.878	26.215

	1	2	3	4	5
1	0	1.8868	3.8207	5.521	6.3544
2	1.8868	4.0554	6.4544	8.6097	9.7073
3	3.8207	6.4544	9.495	12.389	14.087
4	5.521	8.6097	12.389	16.614	19.584
5	6.3545	9.7073	14.087	19.584	26.215

As with the 5-by-5 surface, the z-coordinates calculated using the mathematical model do not correspond with the z-coordinates measured from the fabricated 9-by-9 surface. All of the z-values obtained from the fabricated surface are greater than the z-values of the mathematical model.

One last fabricated surface was analyzed to confirm that the mathematical model did not estimate the movement of the manufactured crust utilizing Becker's unit cell design. The final crust that was a 5-by-5 array constructed from WaterClear 10120 resin. It was believed that the elastic properties that WaterClear 10120 possessed might help the crust to deform more freely. This was not the case in all three displacement tests. The

collected data for displacing the center unit cell by 19.882 millimeters is shown in Table 13. The other measured and calculated displacements can found in Appendix C.

Table 13 – The z-coordinates of a fabricated 5-by-5 surface using WaterClear 10120 resin displaced in the center by 19.882 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	6.976	9.443	6.987	0
2	7.019	13.023	15.493	12.932	7.072
3	9.49	15.661	19.882	15.562	9.677
4	7.048	12.715	15.265	13.039	7.054
5	0	7.123	9.54	7.211	0

	1	2	3	4	5
1	0	5.9945	8.848	5.9945	0
2	5.9945	10.104	13.488	10.104	5.9945
3	8.848	13.488	19.882	13.488	8.848
4	5.9945	10.104	13.488	10.104	5.9945
5	0	5.9945	8.848	5.9945	0

As with the previous fabricated surfaces, the z-displacements measured from the testing apparatus did not coincide with the z-values that were outputted by the mathematical model. All of the z-values from the manufactured model were larger than the displacements in z-direction computed by the mathematical model.

After a closer investigation of the fabricated surface, it appeared that the compliant spherical joints were not being allowed to bend freely. As mentioned briefly in Chapter 4, it seemed that the leaf spring connectors of Becker's unit cell design were hindering the bending movement of the compliant spherical joints. A procedure was constructed to test this theory, which is discussed in the next section.

Hindrance of Bending Caused by the Leaf Springs Test

After the failure of the mathematical model to predict the movement of a formable manufactured surface, it emerged that the leaf spring connectors were not allowing the compliant spherical joints to bend three dimensionally. To confirm this theory, a procedure was designed to test a 1-by-9 array of unit cell on two axes. Figure 81 illustrates the concept of testing a manufactured 1-by-9 array on two axes.

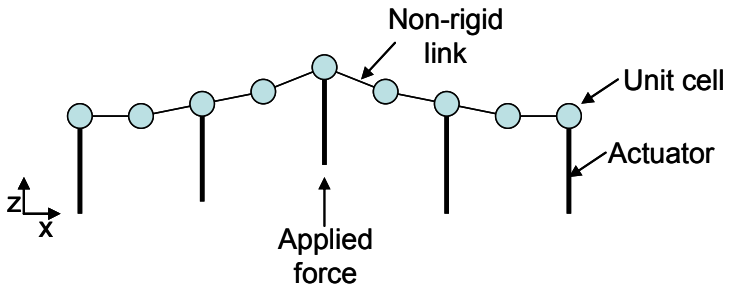


Figure 81 – Illustration of a 1-by-9 array of unit cells being actuated in the center

A comparison of a 1-by-9 array of unit cells to the mathematical model made it possible to determine if the leaf springs were hindering the three dimensional movement of the compliant spherical joint because the 1-by-9 array eliminated the stiffness associated with the leaf springs in the y-direction of Figure 81.

A test stand that was constructed by Sungshik Yim was utilized to test the 1-by-9 array. The rods that were used to imitate the actuators were connected to every other unit cell in the array. Some of the rods were displaced by a different amount in the z-direction and the z-coordinates of the unit cells were recorded. Next, the locations of the moved rods were inputted into the mathematical model to simulate the movement inflicted on the fabricated array and the z-values for the unit cells were calculated. A total of five different rod orientations were examined. The data set shown in Table 14 was acquired from testing a 1-by-9 array that was fixed at the zero points on the ends and the middle unit cell was displaced by 30 millimeters.

Table 14 – Data comparison of a 1-by-9 array of unit cells using the mathematical model (labeled MATLAB positions) and a manufactured array (labeled crust positions)

MATLAB POSITIONS			CRUST POSITIONS		
x	y	z	z1	z2	ave z
0	0	0	0	0	0
18.95	0	8.3974	8.716	8.536	8.626196
37.9	0	15.36	15.943	15.614	15.7785
56.85	0	22.733	23.596	23.109	23.35239
75.8	0	30	30	30	30
94.75	0	22.733	23.725	23.273	23.49891
113.7	0	15.36	16.03	15.725	15.8775
132.65	0	8.3974	8.764	8.597	8.68032
151.6	0	0	0	0	0

As shown in Table 14, the z-coordinate of the constructed 1-by-9 array was measured twice and an average z-position was calculated. A comparison of the z-values in Table 14 showed a greatly improved correlation between the two sets than experienced in the previous section. The z-values obtained from the manufacturing model are slightly higher than the z-coordinates calculated from the mathematical model. This difference could be attributed to measuring error which is depicted in the discrepancies between the two measured z-coordinates. Shown in Figure 82 is a comparison between the unit cell positions calculated by the mathematical model and the shape of the manufactured array when the ends were fixed at the zero position and the center unit cell was displaced 30 millimeters in the z-direction.

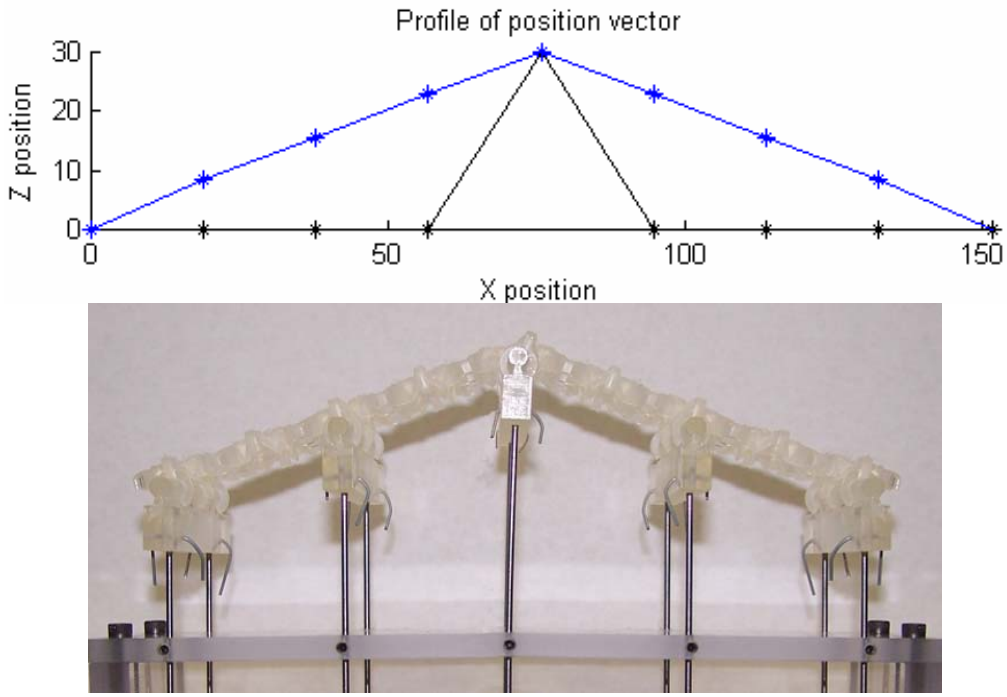


Figure 82 – An illustrative comparison of the mathematical model and a manufactured 1-by-9 array with the ends fixed at zero and the center unit cell displaced 30 mm

As with the data set shown in Table 14, the additional data sets from the remaining rod orientations of the manufactured array correlate remarkably well with the

mathematical model calculations. The data results of the remaining rod orientations are located in Appendix C. The comparison testing of the mathematical model and a fabricated 1-by-9 array of unit cells indicated the hypothesis made in Chapter 4 concerning the increased rigidity of the compliant spherical joint was caused by the leaf springs of Becker's design.

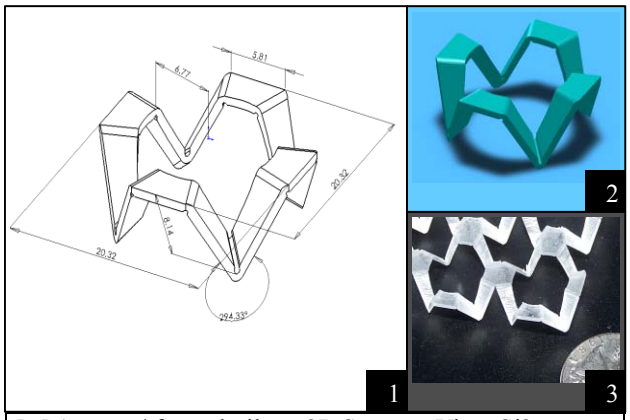
Second Comparison Test

Since the leaf springs of Becker's unit cell design decreased the ability of the compliant spherical joints to bend properly, the four additional unit cell designs discussed in Chapter 4 were used to manufacture further 5-by-5 surfaces and tested against the mathematical model.

The same procedure used in the first comparison test was applied during the examination of the additional unit cell designs. The 5-by-5 surfaces were constructed from the WaterClear 10120 resin and post-processed in the same manner that was adopted in Chapter 7.

First Redesigned Unit Cell

The first unit cell design that was utilized to create a 5-by-5 surface is shown in Figure 83. The increased diameter of the hole did not increase the elasticity of the unit cells. Furthermore, the 5-by-5 crust produced from this unit cell design was not capable of stretching horizontally.



5x5 Array w/ frame built on 3D Systems Viper Si2	
Mode: High Res	Room Temp: 77F
Line Width Compensation: Auto	Room Humidity: 31%
Z Correction: 1	Vat Temp: 25.6C
Resin: WaterClear 10120	Laser Power: 59mW

Figure 83 – Illustration of a redesigned unit cell with an increased center hole and overall unit cell size to increase the elasticity of the unit cell

Without the capability of the crust to stretch and the increased difficulty for the enlarged compliant spherical joints to bend, the fabricated surface, shown in Figure 84, was not analyzed in the testing apparatus to compare it to the mathematical model and it was classified as inadequate.

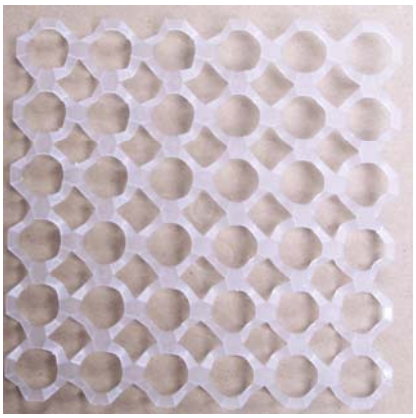
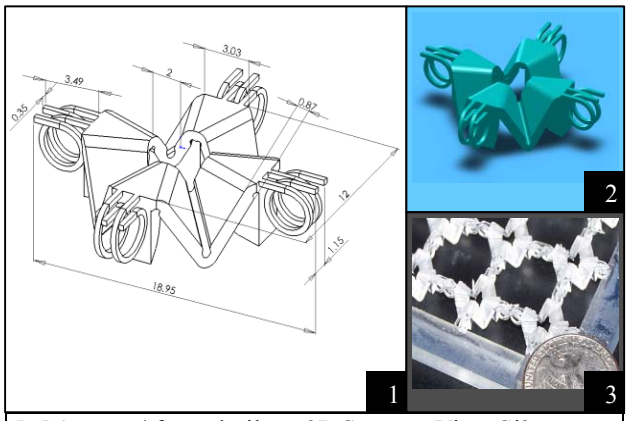


Figure 84 – Fabricated surface utilizing only the compliant spherical joint with a larger diameter through the center

Second Redesigned Unit Cell

The second manufactured 5-by-5 surface was constructed from the unit cell design that incorporated the helixes that replaced the leaf springs of Becker's design. For reference, the helix unit cell is shown in Figure 85.



5x5 Array w/ frame built on 3D Systems Viper Si2	
Mode: High Res	Room Temp: 77F
Line Width Compensation: Auto	Room Humidity: 33%
Z Correction: 1	Vat Temp: 25.9C
Resin: WaterClear 10120	Laser Power: 63mW

Figure 85 – Depiction of the unit cell redesigned with helixes to replacing the leaf springs of Becker's design

As with the previous 5-by-5 surface, this surface was deemed inadequate. As illustrated in Figure 86, the unit cells of the surface could not withstand their own weight. Also, increasing the thickness of the helixes did not appear to affect the ability of the unit cells to endure the weight.

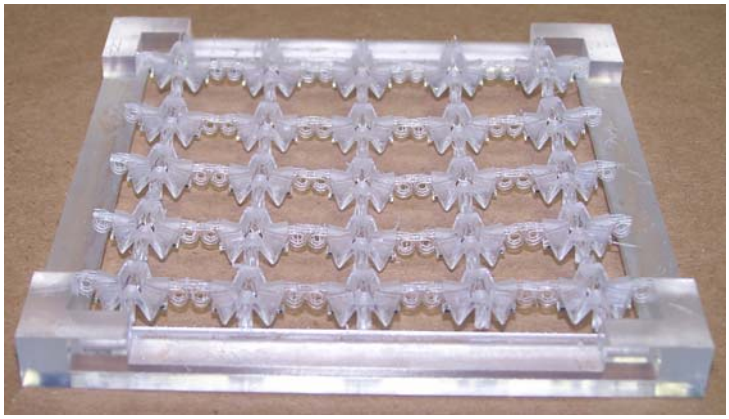
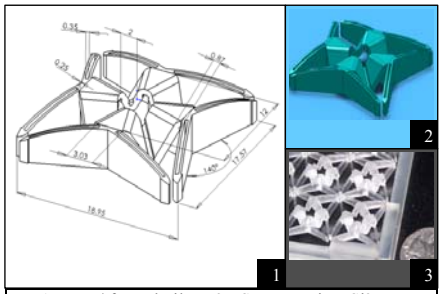


Figure 86 – Illustration of a fabricated surface utilizing helixes bending under its own weight

Third Redesigned Unit Cell

The design concepts of the third fabricated 5-by-5 surface was a proven design even before testing began because it is a modified version of Becker's design. The unit cell utilized in the construction of the third manufactured crust is shown in Figure 87. Unlike the two previous crusts constructed from redesigned unit cells, this crust was capable of stretching horizontally and it was capable of prevailing against sagging under its own weight.



5x5 Array w/ frame built on 3D Systems Viper Si2	
Mode: High Res	Room Temp: 77F
Line Width Compensation: Auto	Room Humidity: 26%
Z Correction: 1	Vat Temp: 25.4C
Resin: WaterClear 10120	Laser Power: 54mW

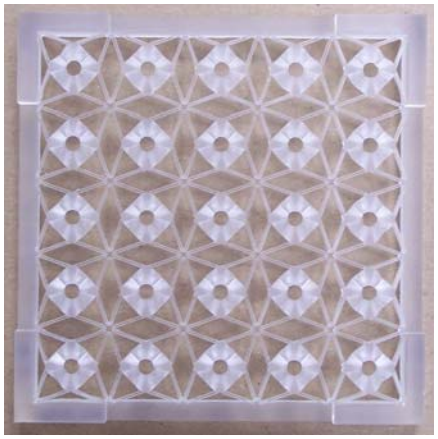


Figure 87 – Unit cell redesign that utilized the design concepts of Becker's unit cell blueprint (left) and the manufactured surface (right)

With a successful preliminary examination of the fabricated crust, the next step completed was a comparison between the z-values of the displacement crust against the z-coordinates calculated by the mathematical model. Table 15 shows the gathered data set of the measured and calculated z-coordinates for the 5-by-5 surface.

Table 15 – The z-coordinates of a fabricated 5-by-5 surface using WaterClear 10120 resin displaced in the center by 8.895 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	2.654	3.949	2.649	0
2	2.684	4.736	6.212	4.749	2.664
3	4.058	5.994	8.895	5.994	4.058
4	2.672	4.753	6.112	4.743	2.669
5	0	2.634	3.931	2.625	0

	1	2	3	4	5
1	0	2.6716	3.9692	2.6716	0
2	2.6716	4.528	6.054	4.528	2.6716
3	3.9692	6.054	8.895	6.054	3.9692
4	2.6716	4.528	6.054	4.528	2.6716
5	0	2.6716	3.9692	2.6716	0

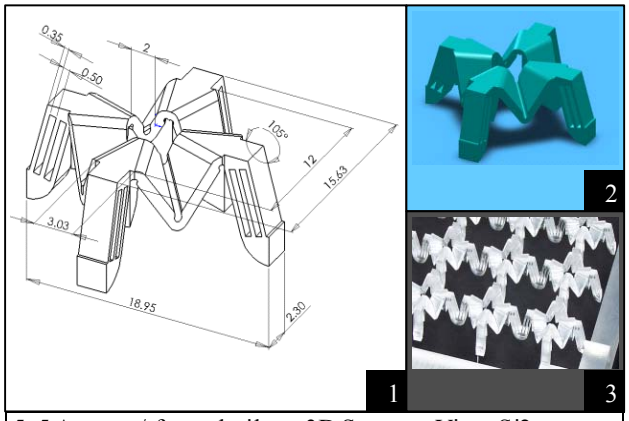
The measured z-coordinates of the fabricated surface correlated closely with the z-values calculated using the mathematical model. The position of the unit cells on the outer edges of the surface differed with the mathematical model by only fractions of a millimeter. The z-values of the constructed unit cells surrounding the centermost unit cell were slightly larger than the z-values of the mathematical model.

The crust was displaced two additional times and the results were similar. The outer ring of the fabricated unit cells was within close proximity to the positions calculated using the mathematical model. Also, the position of the manufactured unit cells that circumvent the middle unit cell were offset slightly from the corresponding unit cells of the mathematical model.

A third displacement was attempted at 30 millimeters but the leaf springs of the unit cells started to bend instead of compliant spherical joint.

Final Redesigned Unit Cell

One last redesigned unit cell was utilized to create a 5-by-5 surface. With the success of the last unit cell design, this design also kept the concepts that Becker employed. For reference, the unit cell is shown in Figure 88 and was discussed previously in Chapter 4.



5x5 Array w/ frame built on 3D Systems Viper Si2	
Mode: High Res	Room Temp: 77F
Line Width Compensation: Auto	Room Humidity: 33%
Z Correction: 1	Vat Temp: 25.9C
Resin: WaterClear 10120	Laser Power: 63mW

Figure 88 – Redesigned unit cell with the leaf spring rotated 90 degrees and utilizing only half of the leaf spring

The correlation between the z-values measured from the physical model and the z-values calculated from the mathematical model is shown in Table 16. As can be seen in the table, all the corresponding z-coordinates had minuscule discrepancies making the comparison and the surface a success.

Table 16 – The z-coordinates of a fabricated 5-by-5 surface using WaterClear 10120 resin displaced in the center by 10.559 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	3.14	4.687	3.131	0
2	3.126	5.361	7.159	5.365	3.127
3	4.691	7.171	10.559	7.163	4.694
4	3.12	5.361	7.167	5.362	3.133
5	0	3.121	4.699	3.146	0

	1	2	3	4	5
1	0	3.165	4.702	3.165	0
2	3.165	5.361	7.181	5.361	3.165
3	4.702	7.181	10.559	7.181	4.702
4	3.165	5.361	7.181	5.361	3.165
5	0	3.165	4.702	3.165	0

It was clear form the acquired data obtained by testing this 5-by-5 surface that the mathematical model could estimate the location of the unit cells accurately when the x-coordinates and the y-coordinates were constrained. The remaining data sets are located in Appendix C and they showed the same correlation between the data sets as Table 16.

Less Constrained Comparison Test

Since it was shown that the mathematical model can accurately estimate the unit cell locations of a fabricated crust constrained in the x-direction and the y-direction, it was desired to examine the predictability of the mathematical model on a less constrained surface.

The procedure for this test followed the process described in the beginning of this chapter with a few alterations. The refinement of the process involved removing 20 of the 25 rods. The locations of the remaining rods were on the four corners of the testing mechanism and the centermost point. The corner and middle unit cells were still constrained in all directions but the rest of the unit cells were not constrained in any direction, as shown in Figure 89.

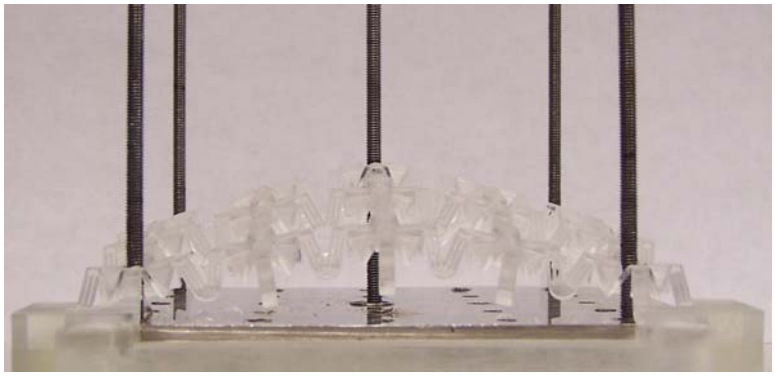


Figure 89 – Depiction of the modified testing structure with rods located at the four corners and in the middle

The surface that was examined was the 5-by-5 crust created from the final redesigned unit cell. Another modification to this comparison test involves the initial guesses of the unit cell positions used in the mathematical model. As mentioned in Chapter 6, the more accurate the starting positions the more likely the mathematical model will output a unique solution. The square pyramid initial z-positions discussed in Chapter 6 were utilized in this evaluation. Only the z-coordinates were measured during the experimentations because of the difficulty to accurately measure the movement of the unit cell in the x-direction and the y-direction. A rod was placed through the center hole of the unit cells until it touched the base plate. Once the rod was contacting the plate, a mark was made on the rod indicating where the unit cell was located. The mark was then measured using the same set of calipers from the previous tests.

The data set shown in Table 17 was acquired from testing a 5-by-5 surface that was fixed at the zero points on the four corners and the middle unit cell was displaced by 24.454 millimeters. The symmetric nature of the problem was used and only a quarter of the unit cell positions were measured.

Table 17 – The z-coordinates of a fabricated 5-by-5 surface using WaterClear 10120 resin displaced in the center by 10.559 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3
1	0	4.2881	5.9103
2	4.2881	11.424	14.356
3	5.9103	14.356	24.454

	1	2	3
1	0	4.272	5.983
2	4.251	11.497	14.404
3	5.883	14.392	24.454

As shown in Table 17, the z-coordinates of the constructed surface showed a good connection to the z-values calculated from the mathematical model. The majority of the unit cell positions of the manufactured surface are lower than the z-coordinates outputted from the mathematical model. The difference in position locations could be attributed to measurement error. Shown in Figure 90 is a visual comparison between the unit cell positions calculated by the mathematical model and the shape of the manufactured crust when the center unit cell was displaced 24.454 millimeters in the z-direction.

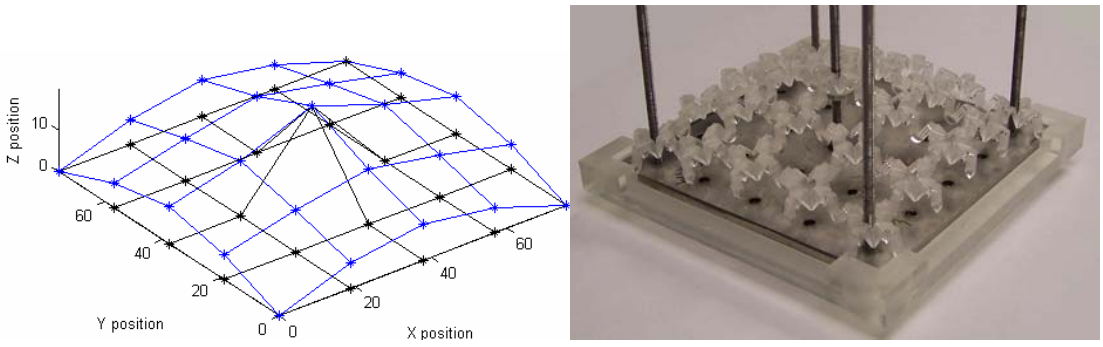


Figure 90 – An illustrative comparison of the mathematical model and a manufactured 1-by-9 array with the ends fixed at zero and the center unit cell displaced 30 mm

As with the data set shown in Table 17, the additional data sets from the remaining z-displacements of the center unit cell of the manufactured crust correlated remarkably well with the mathematical model calculations. The data results of the remaining rod orientations are located in Appendix C. The comparison testing of the z-

coordinates of a less constrained mathematical model and a less constrained manufactured surface was successful. Since the z-values matched well, it is quite probable that the x-coordinates and y-coordinates match.

Comparison Test Summary

In this chapter, three different test procedures were performed on the fabricated crust. Each of these tests had different parameter settings making them unique from one another. The parameter settings inflicted on the manufactured surface were duplicated in the mathematical model allowing for a comparison between unit cell positions measured from the fabricated crust and the unit cell positions calculated from the mathematical model.

The first test procedure discussed in this chapter was a comparison test between the mathematical model and fabricated surfaces utilizing the unit cell design produced by Becker [37]. The constraints imposed on the unit cell during this test were as follows: constrained x-coordinates and y-coordinates on all unit cells, fully constrained corner unit cells, and a displaced centermost unit cell that was also fully constrained. The purpose of this test was to identify if the mathematical model accurately portrayed the movement of the physical model. The test showed that the mathematical model did not predict the movement of the crust constructed of unit cells designed by Becker.

The next test was configured to determine why the mathematical model did not display the same movement as the fabricated surface. In this test, a 1-by-9 array of unit cells was utilized to see if the leaf spring connectors were hindering the compliant spherical joints from fully bending. The 1-by-9 array was displaced in various different forms and the mathematical model mimicked the movement accurately. This proved that the leaf springs were preventing the compliant spherical joints from bending three dimensionally. To correct this problem, the first comparison test was re-administered to surfaces constructed from the redesigned unit cells introduced Chapter 4.

Of the four additional unit cell designs, only two were accurately characterized by the mathematical model. Figure 91 shows the two unit cell designs that were capable of being represented by the mathematical model with Becker's unit cell design.

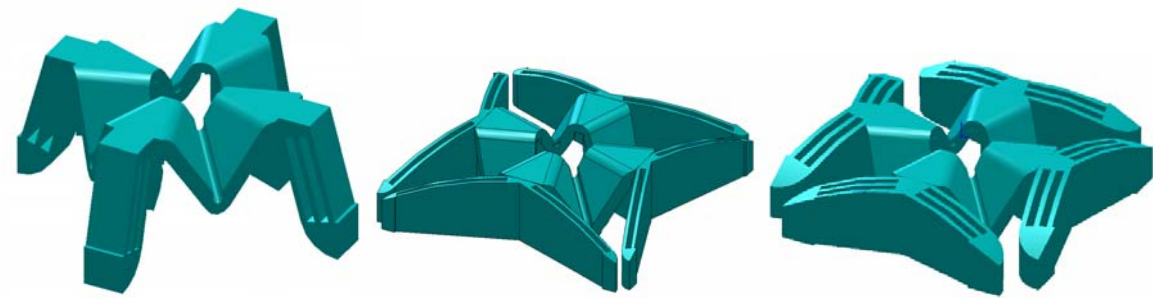


Figure 91 – Redesigned unit cells (left and middle) that produced crust movement replicated by the mathematical model and Becker's unit cell design (right)

The last comparison test was similar to the first comparison test with the exception of the constraints applied to the unit cells. As with the first test, this test fully constrained the four corner unit cells and displaced the middle unit cell in the z-direction whereupon it was fully constrained in all directions. But instead of inducing constraints on the x-coordinates and y-coordinates of the remaining unit cells, no constraints were applied. The comparison between the unit cell positions of the fabricated surface and the mathematical model correlated nicely.

As was evident through the data sets provided in this chapter, the mathematical model did not equate the movement of a fabricated surface when Becker's unit cell design was utilized. However, after the additional unit cell designs were tested under the same conditions, the mathematical model proved to provide comparable unit cell positions.

With a mathematical model capable of forecasting the movement of a fabricated surface, a way to connect the surface to an array of actuators was desired. In the next chapter, different connection possibilities were examined.

CHAPTER 9

CONSTRUCTING A SURFACE FOR THE BED OF NAILS

The last part of this thesis involved the development of a manufactured model of the digital clay device. As described in the first chapter, there are two different digital clay devices being investigated by the Georgia Institute of Technology with a third device being a variant of the bed of nails and formable crust devices. The variant digital clay device includes a computer-interfaced bed of nails array that is bounded by the formable surface. The bed of nails consists of densely packed actuators that are displaced vertically by hydraulic fluid displacement. The actuators can sense pressure changes that are applied to the crust and the height of the crust is adjusted to replicate the force displacement. Figure 92 shows an example of the bed of nails with the crust attached to the actuators.

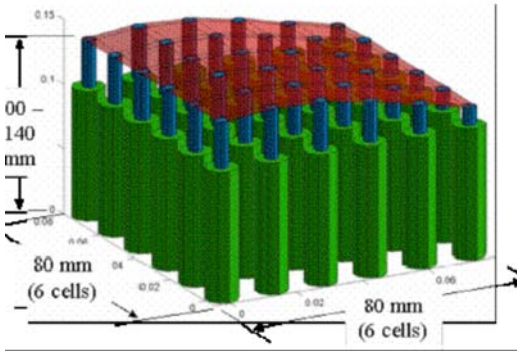


Figure 92 - A display of the variant clay devise that includes a bed of nails and crust

The focus of this chapter was on devising a way to connect the surface to the bed of nails. Several different options were investigated. The first option considered was the placement of the actuator rod on the unit cells. Figure 93 illustrates two placement options.

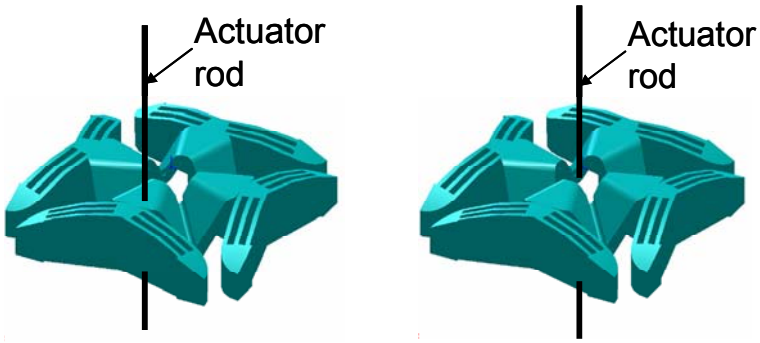


Figure 93 – Actuator rod located through the flat main link of the spherical joint (left) and the actuator rod located through the center of the unit cell (right)

Locating the actuator rod through the center of the unit cell was the more attractive option because it would not have impacted how the unit cells react when displaced. Furthermore, by locating the actuator rods through the flat main link of the spherical joint, the bending capability of the compliant spherical joint would not be maximized and the mathematical model would not accurately represent the movement of a manufactured surface.

The next step was to establish a method to attach the unit cells to the actuator rods. During the initial investigation, three attachment alternatives were reviewed. The first two options were a fabricated part of the unit cell. One of the parts, shown in Figure 94, was designed as an attachment mechanism inside the hole of the unit cell that was used to affix the actuator rod to the cell.

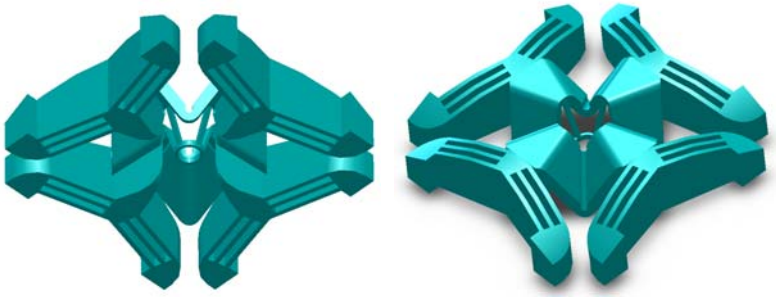


Figure 94 – CAD drawing of a unit cell with an attachment mechanism inside the middle hole of the unit cell

The design incorporated four slanted beams that were adjoined to the edge of the hole of the unit cell by compliant joints and a circular holder that was intended to clasp the actuator rod. The intent of the compliant joints was to minimize the affect the rod holder may have on the ability of the compliant spherical joint to bend. Additionally, four beams were applied to prevent the rod holder from bending or twisting on its own and they were slanted to allow the size of the central hole of the unit cell to fluctuate as the spherical joint deformed. Shown in Figure 95 is a fabricated array of unit cells with the rod holder located in the center of the unit cells.

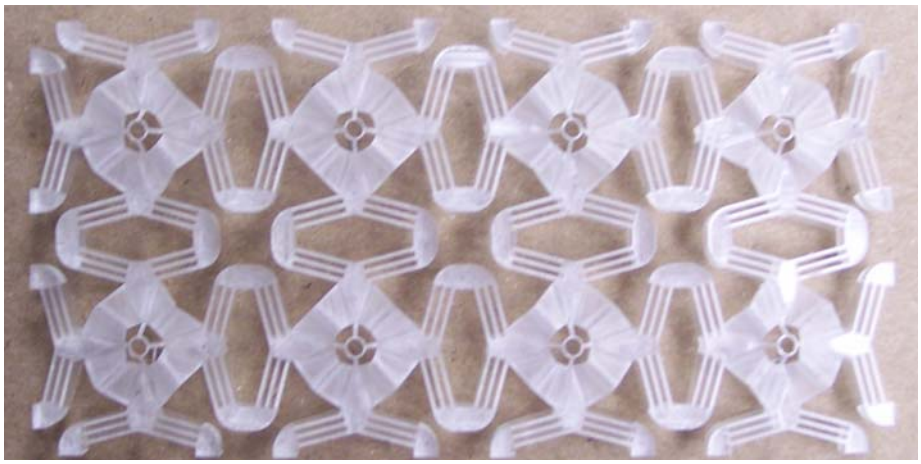


Figure 95 – An array of unit cells fabricated with a rod connecting mechanism located in the center of the unit cells

After testing this rod holder, it was discovered that the needed thickness of the beams hindered the bending motion of the spherical joints. Also, the thinner beams were found to be extremely fragile and broke when the rod was threaded through the hole.

The next adhered rod holder was placed on the bottom of the unit cell and it consisted of several compliant joints to elevate any stiffness it may cause on the ability of the spherical joint to bend. Figure 96 shows the designed unit cell.

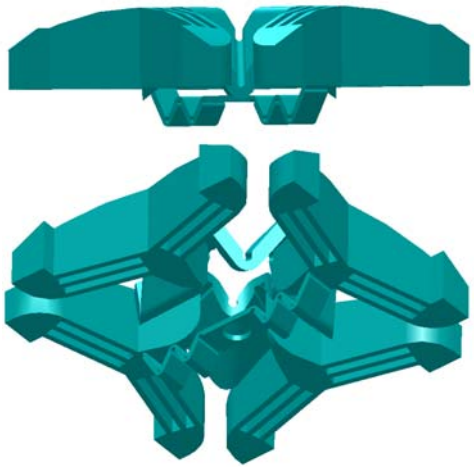


Figure 96 – CAD model of a rod holder that utilized a series of compliant joints

It was desired to have an accordion affect by creating the attachment from compliant joints. The thought was the compliant joints would compress together or disperse apart when the spherical joint deformed. A hole was located at the center of the compliant joints to allow the actuator rod to attach to them. Shown in Figure 97 is a fabricated array of unit cells with the compliant joint rod attachment.

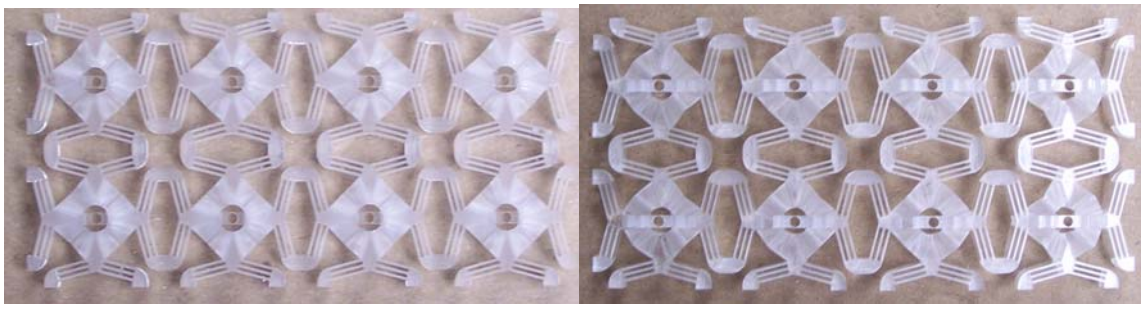


Figure 97 – A top view (left) and bottom view (right) of an array of unit cells fabricated with a rod connecting mechanism consisting of compliant joints located on the bottom of the unit cells

After a preliminary investigation of the fabricated array, it was believed that the compliant joints of the actuator rod holder provided additional stiffness to the spherical joints and the compliant joints bowed instead of compressing. A positive property of this design was that it was not extremely fragile. Unlike the previous rod connecting

mechanism, this one was quite robust and did not break when a rod was placed through the hole located in the middle. However, as illustrated in Figure 98 the mechanism was not able to stabilize the array of unit cells and it distorted under the weight of the unit cells.

Figure 98 – A rod placed through a hole of the compliant joint rod holder mechanism and the weight of the unit cells distorted the mechanism causing the array to be destabilized

The last attachment method investigated was approached differently than the previously discussed methods. The concept of this method was to manipulate the hardware of the actuators instead of fabricating additional parts on the unit cell. This concept was developed from the comparison testing apparatus discussed in Chapter 8. The testing apparatus utilized a washer, a hex nut, and a threaded rod to displace the center unit cell of the fabricated surface in the position z-direction as illustrated in Figure 99.

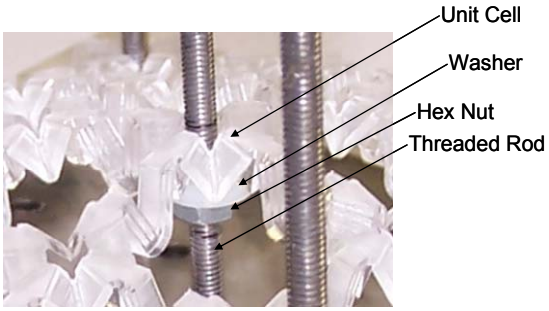


Figure 99 – Illustration of the washer, hex nut, and threaded rod utilized to displace the middle unit cell in the positive z-direction

As shown in Figure 100, the testing apparatus only utilized a method for pushing the surface up. The desire was to have a surface connected to an array of actuators that allowed the actuators to manipulate the surface in both the positive and negative z-direction. To meet this demand, the washer-hex nut combination was utilized on top of the unit cells as well as below them as shown in Figure 100.

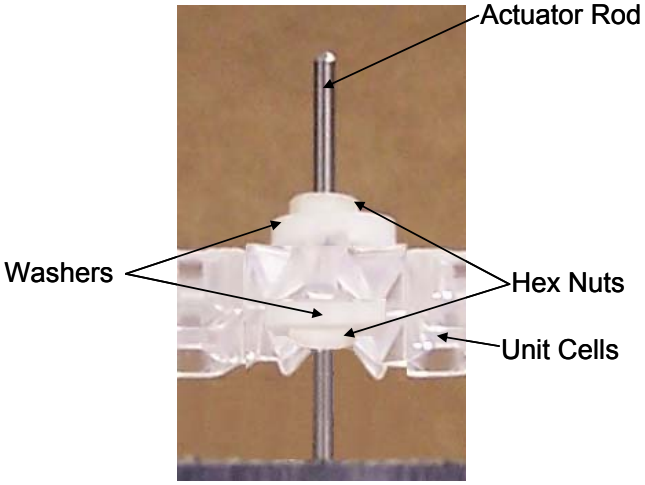


Figure 100 – This figure depicts the configuration of the washer-hex nut combination that secures the fabricated surface to the actuator rod

A test stand was constructed to mimic the positions of the actuators. Rods having diameters of 0.1 inches where used to simulate the rods diameter of the actuators. A die with an inch size of 1-72 was used to thread the rods approximately one inch down from the top of the rods. A hex nut with a thread size in inches of 1-72 was fastened to the bottom of the threading. Next, a washer was placed on top of the secured hex nut and the unit cells rested on top of the washer. The washer was large enough to prevent it from getting trapped inside of the spherical joint and becoming an obstruction to the bending movement of the unit cell. This was identical to the mechanism utilized in Chapter 8. An additional washer-hex nut combination was utilized on the top of the unit cell. This restricted the unit cells from drifting up when a rod was displaced in the position z-

direction. Furthermore, this allowed unit cells to be dragged down by displacing a rod in the negative z-direction. Figure 101 shows a tested crust where the middle unit cell was displaced in the negative z-direction.

Figure 101 – A fabricated surface connected to a test array of rods and displaced in the negative z-direction or pull down in the center

One major advantage to this attachment method was the fact that nothing was fabricated as an additional feature to the unit cell. This eliminated the fragility of the rod connection that was experienced in the previous options. A disadvantage to this method was captured in Figure 101 and has the appearance that the upper washers are not properly attached to the surface. This circumstance occurred because the hex nuts could not be tightened onto the surface. If the hex nuts were clasped tightly against the surface it would have prevented the spherical joints from bending, so an unconstrained fit was used. It was difficult to keep the washers in place when the surface was displaced, which hindered how much movement the crust could experience. This problem could be averted by utilizing washers with an opening slightly larger than the diameter of the rods. Further, the surface would not be able to bend in places where it should not be bending by using a smaller diameter hole. For example, in Figure 101 the ends of the surface are curling upward which should not be the case.

Summary of Connecting the Surface

To create the variant digital clay device described in Chapter 1 and at the beginning of this chapter, the manufactured surface needs to be attached to the bed of nails. Three different fastening options were investigated in this chapter. The first two options consisted of fabricating an extra feature to the unit cell using stereolithography. These extra features were utilized to hold the rods of the actuators to the unit cell. These two options generated an increased burden on the bending ability of the spherical joints. Furthermore, the fabricated features were fragile and broke extremely easily.

The last method exploited the hardware of the actuators to connect the surface to them. This approach was desired because it did not require the development of complex features attached to the unit cell. The surface was secured by threading the top portion of the rods and utilizing a washer-hex bolt combination above and below the surface. The washers needed to be modified slightly to improve the movement of the bound surface. By decreasing the hole-diameter of the washer, the surface would be constrained better by eliminating some of the horizontal and bending movement experienced by the washer. This method proved to be the best of the three options.

CHAPTER 10

CONCLUSIONS

The final chapter in this thesis consists of a review of the research conducted to ensure that the goals presented in Chapter 1 were completed. In addition, this review will specify the achievements and contributions of this work. Lastly, the limitations of the work presented in this thesis are considered.

Review of Work and Goals

The origin of work conducted in this thesis stems from the desire to produce a haptic three dimensional CAD system that allows for a human-computer interface. The work completed in this thesis emanated from a fabricated planar crust that experienced displacement from an applied perpendicular force.

The first fundamental component of this thesis was researching what has already been achieved in the area of haptic systems. The need and desire to create a haptic system was not new. Many different research groups have been exploring different avenues of haptic systems. It was found that the majority of current industrial and research systems fall into the category of having an exoskeleton interface. This interface requires the user to wear or use an external device that produces a resistive force feedback. Unlike these devices, the desire of digital clay is to stimulate the skin to pressure information by touch.

An additional area that was investigated in the area of already completed work involved searching what was already achieved by the research group at the Georgia Institute of Technology. This investigation found that a unit cell design had been extensively developed but a complete comparison between a fabricated surface and a mathematical model had not been achieved. Furthermore, it had become clear that a

mathematical model that represented the stretching ability of the fabricated surface was not investigated.

The first major achievement of this thesis was the further development of unit cell designs that had the capability of stretching horizontally. From the investigation of previous work, it was a concern that the current design provided too much restriction on the deformability of the spherical joints.

The second dominant element of this thesis involved re-formulating a mathematical model that utilized a spring representation of the different components of the unit cell. The compliant spherical joint was represented by a rotational spring and the translational joints were depicted as translational joints. Two different mathematical models were created. The difference between the two models was the number and type of rotational springs applied. Both models employed four translational springs per unit cells and two 180 degree rotational per unit cells but one model incorporated an additional four rotational springs per unit cell. These auxiliary equations were based on the spring representation that was exploited by Nguyen [11].

After the mathematical models were formulated, MATLAB's Optimization Toolbox was used to solve the minimization of the potential energy of the mathematical models by utilizing the MATLAB function lsqnonlin. The two algorithms were tested by two different experiments but the first test was the one that lead to some interesting findings. The first experiment involved a series of displacements of the centermost unit cell in the z-direction while the four corner nodes were fully constrained. This test revealed that a symmetric solution to a symmetric problem solved by the mathematical model was dependent on the number of constraints on the system. After additional testing, it was also identified that the mathematical models could produce a symmetric solution to a symmetric problem of an under constrained system if the accuracy of the initial guess to the locations of the unit cells was increased.

To improve the accuracy of the mathematical models the spring constant values of the fabricated spherical joint and the leaf spring were determined by measuring the displacements of the unit cell from hanging several different masses from various locations on the unit cells. The amount of force exerted on the unit cell was graphed against the amount of displacement that was experienced. The graph produced was linear indicating that the unit cell was still in the elastic stage. Furthermore, the slope of the line was utilized to obtain the spring constants. The measured spring constants were then verified as acceptable by data obtained from a FEA model simulating the same test procedures.

The third component of this thesis was an experimental investigation into the functionality of the mathematical model compared to a fabricated surface. Initially two different resins were applied to manufacture a series of 5-by-5 surfaces and a 9-by-9 surface. The constraints induced on the constructed models were equivalent to the constraints inflicted to the mathematical models. Initial testing did not demonstrate the desired results. The displacement of the mathematical model did not mimic the movement of the fabricated surfaces. After testing the surfaces fabricated from different resins and testing a different sized surface, it was determined that Becker's unit cell design might be inadequate. To eliminate the influence of the leaf springs, a 1-by-9 array of unit cells was constructed and tested against the mathematical model. The displacement results of the mathematical model showed good correlation with the experimental results that were measured. This concluded that Becker's unit cell design was defective because the leaf springs were not allowing the spherical joints to deform three dimensionally.

The four redesigned unit cells were used to create 5-by-5 arrays of unit cells. Displacement of the surfaces showed that two of the four designs produced good correlation with the calculated data from the mathematical models. Additionally, the two

models achieved good correlation with the mathematical model when fewer constraints were introduced to the system.

Evaluation of Goals

In Chapter 1, a number of goals and contributions were proposed. In this section, those goals are restated along with a discussion on the progress made toward achieving those goals.

First Goal

As stated in Chapter 1, the main goal of this thesis was to formulate a computational algorithm that calculates the position of a crust based on material properties and constraints induced on the system.

The crux of this goal was achieved in Chapters 5, 6, and 7. In Chapter 5, mathematical models were produced by utilizing the representation of springs for the joint of the unit cell. In Chapter 6, algorithms were developed to minimize the potential energy produced by the springs represented in the mathematical model. The lsqnonlin function of MATLAB was employed to solve the least squares problem associated with finding the unit cell positions by minimizing the potential energy of the crust. The algorithms were tested by simulating a crust being pushed up and by simulating a crust being pushed up then dragged. The preliminary results of the mathematical model appeared reasonable because the links between the unit cells were slightly stretched, not compressed. Lastly, Chapter 7 described how experimental procedures were utilized to measure the spring constants of the fabricated unit cells. To verify the experimental results, the spring constants were calculated theoretically by creating FEA models in ANSYS and applying the same constraints that were introduced to the constructed unit cells.

Through completed research done in Chapter 5 through Chapter 7, this goal was completed and a mathematical model capable of calculating the position of a crust was created.

Second Goal

The second goal, as stated in Chapter one, was to fabricate a formable crust for the digital clay haptic device. It can be observed from literature that the fabrication of a formable crust has been explored experimentally. No attempt has been made to compare mathematical models to the manufactured crust.

An experimental procedure was completed in Chapter 8 to compare the reaction of a fabricated surface to that of the mathematical model. The first experiments were disappointing because the data sets of the mathematical model and the fabricated surface did not correlate. But after investigating four alternative crust designs, the movement of two of the surfaces was imitated by the mathematical model. After some of the constraints were eliminated, the unit cell positions of the mathematical model correlated well with the unit cell positions of the manufactured surface.

This goal was achieved by showing good correlation between the unit cell positions of the mathematical model and of the fabricated surface in the experiments completed in Chapter 8.

Third Goal

The last objective stated in Chapter 1 was to expand on the development of the formable crust by devising a way to integrate the surface with the bed of nails.

In Chapter 9, three different options were investigated for attaching the surface to the bed of nails. It was found that fabricating additional features to the unit cells was not the ideal way to affix the surface to the rods of the actuators. The reasons included hindering the ability of the spherical joint to deform and the features being too fragile to

withstand the forces induced on them by the rods. The last technique investigated was utilizing the hardware of the actuators to attach the surface to the rods. Some mock rods were threaded using a 1-72 inch die and a hex nut was threaded to the rod. A washer was then placed on top of the nut followed by a fabricated surface. The washer-hex nut combination was applied to hold the crust in place. An additional washer and hex nut were placed on top of the crust to give more support and control when deforming the surface. Preliminary testing of this technique was positive.

From the research completed in Chapter 9, it was concluded that the best approach to affixing the surface to the bed of nails was by threading the top portion of the actuator rods and placing hex nuts with washers to hold the surface in place.

Limitations

The objective of this research was to create a mathematical model for predicting the movement of a fabricated surface. The mathematical model that was developed is a very simple method utilizing potential energy stored in springs. The mathematical model capable of mimicking the movement of a manufactured crust was created utilizing this principle. For surfaces composed of large arrays of unit cells this mathematical model will be insufficient because it will take an extremely long time to compute the new unit cell locations. Additionally, it was shown that the system either needs to have a certain degree of constraint inflicted upon it or the initial positions of the unit cells need to be quite accurate.

To further increase the accuracy of the mathematical model, the mechanical properties of resins are needed. As a result of not having the exact mechanical properties, a range had to be applied to the mechanical properties of the resin for the purpose of calculating the spring constants of the unit cells.

The experimental measurements were obtained manually, so some inaccurate measurements are expected. The effects of the faulty measurements have not been

analyzed in this research but some discrepancies in unit cell positions in the comparison of the manufactured surface and mathematical model were expected and observed.

Future Work

The primary direction in this thesis involved the investigation of a planar surface that was displaced by inflicting a force perpendicular to the surface. A mathematical model was developed to calculate the unit cell positions of a crust that correlated with a manufactured surface. The models were only displaced in one axial direction. For this device to take on the characteristics of real clay, an investigation into multi-axial displacement will need to be investigated. This can be achieved in the same manner as the research completed in this thesis. Additionally, the fabricated surface investigated in this thesis was a planar surface. An examination into a more complex surface is the next step into creating a digital clay device that is comparable to real clay. This shape could be in the form of a square or sphere that utilizes the same unit cells that were evaluated in this thesis. By adding additional springs to the mathematical model, it may improve the predictability of unit cell placement when multi-directional forces are applied to a manufactured surface.

The current algorithms used to solve the mathematical models were designed with the purpose of calculating the unit cell positions by minimizing the potential energy of the surface. The MATLAB function lsqnonlin was employed to solve the algorithms. This system appears to handle smaller crust sizes quite well. As was demonstrated experimentally, as the size of the surface increases, so does the amount of time required to solving the algorithms. Further experimental testing should investigate the performance of the mathematical model with the intent of decreasing the computational time.

In addition, testing should be undertaken to establish the effect of affixing the manufactured surface to an array of actuators. A comparison between simulation testing

of the variant digital clay device and the mathematical model would provide information into whether the recommended fastening method changes the deformability of the fabricated surface.

APPENDIX A

MATLAB CODES

Main Mesh Generating Code

```
%% Deformable Crust Simulation

clear all;
clc
%=============INPUTS=================================================
================
global ncells;
global link_len;
global xcell_var;
global xcell_initial;
global xcell;
global jacob_map jacob_map2 jacobian__;

%link distance between original link locations
link_len = 18.95; %units=mm
%size of the crust
ncells=[5,5]; %needs to be odd numbers for the matrix

% Coordinates for crust, initial conditions, creates xcell
num_cells = ncells(1)*ncells(2);
xcell = zeros (num_cells, 3);
for k = 1: ncells(1)
  for kk = 1: ncells(2)
    xcell((k-1)*ncells(2)+kk,1) = (kk-1)*link_len;
    xcell((k-1)*ncells(2)+kk,2) = (k-1)*link_len;
  end
end

xcell_initial=xcell; %storing the intital guess of all the coordinates

% Specify which coordinates are variables (=1) and which are fixed (=0).
% In next block of code, 1's get replaced by indices into Xvec.
xcell_var = ones (num_cells, 3);
col = fix (0.75*ncells(2));
row = fix (0.75*ncells(1));

% Fix x coord of cell below (bottom) cell being controlled.
% Fix y coord of cell to the far left of cell being controlled.
% xcell_var(col, 1) = 0;
% xcell_var((row-1)*ncells(2)+1, 2) = 0;
%fixed coord of cells for 9x9
% xcell_var(1, 1)=0;
% xcell_var(1, 2)=0;
% xcell_var(1, 3)=0;
```

```
% xcell(1, 3)=5.844;
% xcell_var(9, 1)=0;
% xcell_var(9, 2)=0;
% xcell_var(9, 3)=0;
% xcell(9, 3)=5.844;
% xcell_var(73, 1)=0;
% xcell_var(73, 2)=0;
% xcell_var(73, 3)=0;
% xcell(73, 3)=5.844;
% xcell_var(81, 1)=0;
% xcell_var(81, 2)=0;
% xcell_var(81, 3)=0;
% xcell(81, 3)=5.844;

%fixed coord of cells for 5x5
xcell_var(1, 1)=0;
xcell_var(1, 2)=0;
xcell_var(1, 3)=0;
xcell(1, 3)=0;
xcell_var(5, 1)=0;
xcell_var(5, 2)=0;
xcell_var(5, 3)=0;
xcell(5, 3)=0;
xcell_var(21, 1)=0;
xcell_var(21, 2)=0;
xcell_var(21, 3)=0;
xcell(21, 3)=0;
xcell_var(25, 1)=0;
xcell_var(25, 2)=0;
xcell_var(25, 3)=0;
xcell(25, 3)=0;


% % Fix y, z coords of cells along bottom row
for k = 1: ncells(2)*ncells(1)
   xcell_var(k,2) = 0;
   xcell_var(k,1) = 0;
end
% for k = 1: ncells(1)
%    xcell_var((k-1)*ncells(2)+1,3) = 0;
% end

%Fix cell being controlled
xcell(round((ncells(1)*ncells(2))/2),3)=18.143;
xcell_var(round((ncells(1)*ncells(2))/2),3) = 0;
xcell_var(round((ncells(1)*ncells(2))/2),2) = 0;
xcell_var(round((ncells(1)*ncells(2))/2),1) = 0;

% Rearrange coordinates into X vector.  Don't put constants in X.
%1's from xcell_var get replaced by indices into Xvec.
xcnt = 1;
for k = 1: ncells(1)
  for kk = 1: ncells(2)
     index = (k-1)*ncells(2)+kk;
     for x = 1: 3
```

```
        if xcell_var(index, x) ~= 0
          xvec(xcnt) = xcell(index,x);
          xcell_var(index, x) = xcnt;
          xcnt = xcnt + 1;
        end
      end
    end
end


% Determine Jacobian Map -- array of indices that record relationships
%  between variables (Xvec) and entries in the Jacobian.
jacob_map = zeros (ncells(1)*(ncells(2)-1) + ncells(2)*(ncells(1)-1), length(xvec)); % should be
(12x17) for this example
sjac = size (jacob_map);
jacob_map2 = zeros (sjac(1), sjac(2));
jacobian__ = zeros (sjac(1), sjac(2));

for k = 1: ncells(1)
  for kk = 1: ncells(2)-1
    vind = (k-1)*ncells(2)+kk;
    jind = (k-1)*(ncells(2)-1) + kk;
    for x = 1: 3
      if xcell_var(vind, x) ~= 0
        jacob_map(jind, xcell_var(vind, x)) = xcell_var(vind, x);
        jacob_map2(jind, xcell_var(vind, x)) = (vind+1)*10 + x;
      end
    end
    for x = 1: 3
      if xcell_var(vind+1, x) ~= 0
        jacob_map(jind, xcell_var(vind+1, x)) = xcell_var(vind+1, x);
        jacob_map2(jind, xcell_var(vind+1, x)) = vind*10 + x;
      end
    end
  end
end
jind_old = jind;

for k = 1: ncells(1)-1
  for kk = 1: ncells(2)
    vind = (k-1)*ncells(2)+kk;
    jind = (k-1)*(ncells(2)) + kk + jind_old;
    vind2 = k*ncells(2)+kk;
    for x = 1: 3
      if xcell_var(vind, x) ~= 0
        jacob_map(jind, xcell_var(vind, x)) = xcell_var(vind, x);
        jacob_map2(jind, xcell_var(vind, x)) = vind2*10 + x;
      end
    end
    for x = 1: 3
      if xcell_var(vind2, x) ~= 0
        jacob_map(jind, xcell_var(vind2, x)) = xcell_var(vind2, x);
        jacob_map2(jind, xcell_var(vind2, x)) = vind*10 + x;
      end
    end
```

```
  end
end

jacob_map;

figure;
graph_crust (ncells, xcell, 'k*-');
% Solve for joint locations.
ffun = 'potential_energy_try4b';
options = optimset ('Display', 'iter', 'TolFun', 0.002, 'TolX', 0.002);
%options = optimset ('Display', 'iter', 'MaxIter', 1e5, 'MaxFunEvals', 1e12, 'TolFun',
0.0001,'TolX',0.0001);
[x,resnorm,fval,exitflag,output] = lsqnonlin(ffun, xvec, [], [], options);
%[x,fval,exitflag,output] = fminunc(ffun,xvec,options);
%[x,fval,exitflag,output] = fminsearch(ffun,xvec,options);

graph_crust (ncells, xcell, 'b*-');

fval
exitflag
```

Graphing Function Code

This is the MATLAB code for the function called graph_crust. This function is used to graph the horizontal and vertical lines for the grid that is displayed to represent the connections between the unit cells.

```
%=====================================================================
function graph_crust (ncells, xcell, color)

holding = 0;

%drawing the horizontal lines for the graph that connects two points
counter35=0;
for it35=1:ncells(1) % the numbers of rows
   for it36= 1:ncells(2)-1 %2  1 less line than the total number of point per rows b/c # of
lines connecting two points
      plot3([xcell(it36+counter35,1);xcell(it36+1+counter35,1)],...
         [xcell(it36+counter35,2);xcell(it36+1+counter35,2)],...
         [xcell(it36+counter35,3);xcell(it36+1+counter35,3)], color);
       if holding == 0
         holding = 1;
         hold on;
       end
   end
   counter35=counter35+ncells(2);
end
```

```
%drawing the vertical lines for the graph that connects two points
counter36=0;
for it38= 1: ncells(1)-1% numbers of column of lines
   for it37=1:ncells(2)
      plot3([xcell(it37+counter36,1);xcell(it37+ncells(2)+counter36,1)],...
         [xcell(it37+counter36,2);xcell(it37+ncells(2)+counter36,2)],...
         [xcell(it37+counter36,3);xcell(it37+ncells(2)+counter36,3)], color);
   end
   counter36=counter36+ncells(2); %3
end

title('Profile of position vector')
xlabel('X position')
ylabel('Y position')
zlabel('Z position')
axis equal
```

Potential Energy Function

This is the MATLAB code for the function called potential_energy_try5. This function is used to calculate the potential energy of the surface as a whole.

```
%The potential energy function

%Program works using two rotational springs in a 180 degree config
%along with four springs in a 90 degree config

%Calculates the potential energy of each spring summed at each node
%This calculation can take some springs into account more than once

%clear all
%clc

function [potential_energy] = potential_energy_try5(xv,xcell)

 global ncells;
 global link_len;
 global xcell_var;
 global xcell_initial;
 global xcell;
 global jacob_map jacob_map2 jacobian__;

%Spring constants
%k_trans is spring constant for the translational springs
%k_rotat is spring constant for the rotational/torsion springs
k_trans = 1993;
k_rotat = .0775;
```

```
%  ncells = [3,3];
%  link_len = 10;
%  xv = 0;
%
%  xcell = [0         0         0;
%      9.5269    0.7744   -0.9392;
%     19.3224    0.3430   -3;
%      0.0       9.5602    1.1292;
%      9.7155    9.8242    -1.0;
%     19.1769    9.8881    0.9258;
%      0.0      19.5543    1.0;
%      9.6926   19.1189    1.0000;
%     19.2082   19.1168    3.0];

% Specify which coordinates are variables (=1) and which are fixed (=0).
% xcell_var = [0 0 0;
%    1 0 1;
%    1 1 0;
%    0 1 1;  %second row
%    1 1 0;
%    1 1 1;
%    0 1 0;  %third row
%    1 1 1;
%    1 1 0];   % 19 vars

xcell_size=size(xcell);

% Rearrange coordinates from X vector.
 xcnt = 1;
 for k = 1: ncells(1)
    for kk = 1: ncells(2)
       index = (k-1)*ncells(2)+kk;
       for x = 1: 3
          if xcell_var(index, x) ~= 0
             xcell(index,x) = xv(xcnt);
             xcnt = xcnt + 1;
          end
       end
    end
 end

%pe = zeros(ncells(2)*ncells(1),1);
potential_energy = 0;
%Calculates the size of the potential engergy matrix
vcnt = 1;
%calculates the potential energy for each node in the crust
%if statements are needed to ensure the edge and corner nodes are
%treated as "special" nodes and only the springs that are there will
%be calculated.
for k = 1:ncells(1) %y direction
    for kk = 1:ncells(2) %x direction
        index = (k-1)*ncells(2)+kk; %locates which point in the xcell matrix will be looked at for
calculating the potential energy at that point
        if index+1 <= 0 | index+1 > xcell_size(1) | kk == ncells(2)
```

```
        pe1 = 0;
% * represents spring location @ is node point
%   *
%  -@-
%   |
      else
         pe1 = sqrt(0.5)*sqrt(k_trans)*abs((sqrt((xcell(index+1,1) - xcell(index,1))^2 + ...
            (xcell(index+1,3) - xcell(index,3))^2))-link_len);
      end
      if index+ncells(2) <= 0 | index+ncells(2) > xcell_size(1)
         pe2 = 0;
% * represents spring location @ is node point
%   |
%  -@*
%   |
      else
         pe2 = sqrt(0.5)*sqrt(k_trans)*abs((sqrt((xcell(index+ncells(2),2) - xcell(index,2))^2 + ...
            (xcell(index+ncells(2),3) - xcell(index,3))^2))-link_len);
      end
      if index-ncells(2) <= 0 | index-ncells(2) > xcell_size(1)
         pe3 = 0;
% * represents spring location @ is node point
%   |
%  -@-
%   *
      else
         pe3 = sqrt(0.5)*sqrt(k_trans)*abs((sqrt((xcell(index-ncells(2),2) - xcell(index,2))^2 + ...
            (xcell(index-ncells(2),3) - xcell(index,3))^2))-link_len);
      end
      if index-1 <= 0 | index-1 > xcell_size(1) | kk == 1
         pe4 = 0;
% * represents spring location @ is node point
%   |
%  *@-
%   |
      else
         pe4 = sqrt(0.5)*sqrt(k_trans)*abs((sqrt((xcell(index-1,1) - xcell(index,1))^2 + ...
            (xcell(index-1,3) - xcell(index,3))^2))-link_len);
      end
      if index+1 <= 0 | index+1 > xcell_size(1) | index-1 <= 0 |...
            index-1 > xcell_size(1) | kk == ncells(2) | kk == 1
         pe5 = 0;
%Represents the rotational spring in the x - direction in xz plane
      else
         pe5 = sqrt(0.5)*sqrt(k_rotat)*(-pi + abs(-pi + ...
            (atan2(xcell(index+1,3) - xcell(index,3),  ...
            xcell(index+1,1) - xcell(index,1))) - atan2(xcell(index,3)...
            - xcell(index-1,3), xcell(index,1)-xcell(index-1,1))));
      end
      if index-ncells(2) <= 0 | index-ncells(2) > xcell_size(1) |...
            index+ncells(2) <= 0 | index+ncells(2) > xcell_size(1)
         pe6 = 0;
%Represents the rotational spring in the y - direction in yz plane
      else
          pe6 = sqrt(0.5)*sqrt(k_rotat)*(-pi + abs(-pi + ...
             (atan2(xcell(index+ncells(2),3) - xcell(index,3), ...
```

```
            xcell(index+ncells(2),2) - xcell(index,2))) - atan2...
            (xcell(index,3) - xcell(index-ncells(2),3), xcell(index,2) ...
            - xcell(index-ncells(2),2)))) ;
        end
        if kk == ncells(2) | index+ncells(2) <= 0 | index+ncells(2) > xcell_size(1)
           pe7 = 0;
        else
           pe7 = sqrt(0.5)*sqrt(k_rotat)*(abs(...
              (atan2(xcell(index+1,2) - xcell(index,2), xcell(index+1,1)...
              - xcell(index,1))) + atan2(xcell(index+ncells(2),1)...
              - xcell(index,1), xcell(index+ncells(2),2)-xcell(index,2))));
        end
        if kk == ncells(2) | index-ncells(2) <= 0 | index-ncells(2) > xcell_size(1)
           pe8 = 0;
        else
           pe8 = sqrt(0.5)*sqrt(k_rotat)*(abs(...
              (atan2(xcell(index+1,2) - xcell(index,2), xcell(index+1,1)...
              - xcell(index,1))) + atan2(xcell(index,1)...
              - xcell(index-ncells(2),1), xcell(index,2)-xcell(index-ncells(2),2))));
        end
        if kk == 1 | index-ncells(2) <= 0 | index-ncells(2) > xcell_size(1)
           pe9 = 0;
        else
           pe9 = sqrt(0.5)*sqrt(k_rotat)*(abs(...
              (atan2(xcell(index,2) - xcell(index-1,2), xcell(index,1)...
              - xcell(index-1,1))) + atan2(xcell(index,1)...
              - xcell(index-ncells(2),1), xcell(index,2)-xcell(index-ncells(2),2))));
        end
        if kk == 1 | index+ncells(2) <= 0 | index+ncells(2) > xcell_size(1)
           pe10 = 0;
        else
           pe10 = sqrt(0.5)*sqrt(k_rotat)*(abs(...
              (atan2(xcell(index,2) - xcell(index-1,2), xcell(index,1)...
              - xcell(index-1,1))) + atan2(xcell(index+ncells(2),1)...
              - xcell(index,1), xcell(index+ncells(2),2)-xcell(index,2))));
        end

%       potential_energy = potential_energy + pe1 + pe2 + pe3 + pe4 + pe5 + pe6;
        potential_energy(vcnt) = abs(pe1) + abs(pe2) + abs(pe3) + abs(pe4)...
           + abs(pe5) + abs(pe6) + abs(pe7) + abs(pe8) + abs(pe9) + abs(pe10);
        vcnt = vcnt + 1;
     end
end

potential_energy;
```

APPENDIX B

EXPERIMENTAL DATA

Experimental Data for Leaf Springs

Table 18 – Experimental data acquired for calculating the spring constant for the leaf spring

Weight (g)	Displacement	Force	Spring constant
	Specimen #1		
52.4	0.2535	0.514044	2027.786982
78.65	0.395	0.7715565	1953.307595
104.9	0.522	1.029069	1971.396552
132.8	0.6675	1.302768	1951.71236
160.7	0.803	1.576467	1963.221669
	Specimen #2		
52.5	0.263	0.515025	1958.269962
78.3	0.3810	0.768123	2016.070866
104.1	0.5105	1.021221	2000.432909
130.3	0.6640	1.278243	1925.064759
156.5	0.802	1.535265	1914.295511
	Specimen #3		
52.2	0.266	0.512082	1925.120301
78.55	0.3940	0.7705755	1955.775381
104.9	0.522	1.029069	1971.396552
130.25	0.6635	1.2777525	1925.776187
155.6	0.7705	1.526436	1981.097988
	Specimen #4		
52.4	0.261	0.514044	1969.517241
78.65	0.3830	0.7715565	2014.507833
104.9	0.522	1.029069	1971.396552
132.8	0.6615	1.302768	1969.414966
160.7	0.821	1.576467	1920.17905
	Specimen #5		
52.5	0.2615	0.515025	1969.502868
78.3	0.3903	0.768123	1968.284433
104.1	0.529	1.021221	1930.47448
130.3	0.6475	1.278243	1974.120463
156.5	0.796	1.535265	1928.724874
	Specimen #6		
52.2	0.264	0.512082	1939.704545
78.55	0.3910	0.7705755	1970.78133
104.9	0.528	1.029069	1948.994318
130.25	0.6450	1.2777525	1981.011628
155.6	0.781	1.526436	1954.463508
	Specimen #7		
52.4	0.2605	0.514044	1973.297505
78.65	0.3950	0.7715565	1953.307595
104.9	0.534	1.029069	1927.095506
132.8	0.6640	1.302768	1962
160.7	0.807	1.576467	1953.490706
	Specimen #8		
52.5	0.262	0.515025	1965.744275
78.3	0.3850	0.768123	1995.124675
104.1	0.511	1.021221	1998.475538
130.3	0.6445	1.278243	1983.309542
156.5	0.776	1.535265	1978.434278
	Specimen #9		
52.2	0.2615	0.512082	1958.248566
78.55	0.3940	0.7705755	1955.775381
104.9	0.524	1.029069	1963.872137
130.25	0.6670	1.2777525	1915.670915
155.6	0.788	1.526436	1937.101523

Experimental Data for Spherical Joints

Table 19 – Experimental data acquired for calculating the spring constant for the spherical joints

angle (degrees)	angle (rad)	displacement	Force	spring constant	regression
24.5	0.427605667	0.004318006	3.626757	839914.7135	0.033134414
30.1	0.525344105	0.005492465	4.140801	753905.7065	0.040707994
12.01	0.209614043	0.002015702	1.576467	782093.3244	0.016242625
8.09	0.141197136	0.001346805	1.062423	788846.9152	0.010941119
15.15	0.264417382	0.002565424	2.083644	812202.6566	0.020489239
21.35	0.372627795	0.003703677	3.112713	840438.6233	0.028874275
29.29	0.511206938	0.005314948	4.140801	779085.9272	0.039612529
7.27	0.126885437	0.001208733	1.062423	878955.6662	0.00983213
15.05	0.262672052	0.002547683	2.083644	817858.4799	0.020353997
21.13	0.368788071	0.003661799	3.112713	850050.1733	0.028576741
29.54	0.515570261	0.005369433	4.140801	771180.3503	0.039950636
7.39	0.128979832	0.001228906	1.062423	864527.4135	0.009994421
14.96	0.261101256	0.00253173	2.083644	823011.8662	0.020232279
29.33	0.51190507	0.005323647	4.140801	777812.7882	0.039666626
7.72	0.134739418	0.001284438	1.062423	827149.9038	0.010440721
14.6	0.254818071	0.002468053	2.083644	844246.1759	0.019745406

APPENDIX C

COMPARISON DATA SETS

First Comparison Test Data Sets

Table 20 – The z-coordinates of a fabricated 5-by-5 surface using SL 5510 resin displaced in the center by 20.358 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	8.374	9.629	8.482	0
2	8.223	14.5	16.515	13.723	8.473
3	9.889	17.045	20.358	16.3	10.044
4	8.334	14.111	16.563	14.305	8.424
5	0	8.403	9.788	8.354	0

	1	2	3	4	5
1	0	6.1363	9.0548	6.1363	0
2	6.1363	10.352	13.815	10.352	6.1363
3	9.0548	13.815	20.385	13.815	9.0548
4	6.1363	10.352	13.815	10.352	6.1363
5	0	6.1363	9.0548	6.1363	0

Table 21 – A quarter of the z-coordinates of a fabricated 9-by-9 surface using SL 5510 resin displaced in the center by 17.96 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	1.43	2.192	2.537	3.416
2	1.673	2.481	2.987	4.312	4.381
3	2.23	2.927	4.719	5.529	6.451
4	2.885	3.673	5.838	6.522	7.98
5	3.505	4.419	6.502	7.932	8.511

	1	2	3	4	5
1	0	1.4552	2.5025	3.3305	3.7179
2	1.4552	2.2479	3.2008	4.0204	4.4356
3	2.5025	3.2008	4.0924	4.9709	5.4846
4	3.3305	4.0204	4.9709	6.0581	6.8665
5	3.7179	4.4356	5.4846	6.8665	8.511

Table 22 – A quarter of the z-coordinates of a fabricated 9-by-9 surface using SL 5510 resin displaced in the center by 17.96 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	2.72	5.069	6.278	7.287
2	2.786	4.63	7.566	9.006	10.182
3	5.063	8.124	10.964	12.976	13.383
4	6.527	9.204	12.697	15.316	16.685
5	7.324	9.946	13.408	16.385	17.96

	1	2	3	4	5
1	0	2.9612	5.0894	6.7666	7.5652
2	2.9612	4.5892	6.5566	8.2567	9.118
3	5.0894	6.5566	8.4353	10.297	11.384
4	6.7666	8.2567	10.297	12.63	14.404
5	7.5652	9.118	11.384	14.404	17.96

Table 23 – The z-coordinates of a fabricated 5-by-5 surface using WaterClear 10120 resin displaced in the center by 9.616 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	2.718	3.686	2.661	0
2	2.684	5.27	7.498	5.28	2.712
3	3.699	7.542	9.616	7.596	3.716
4	2.711	5.312	7.501	5.328	2.688
5	0	2.691	3.651	2.7	0

	1	2	3	4	5
1	0	2.8915	4.2934	2.8915	0
2	2.8915	4.896	6.5444	4.896	2.8915
3	4.2934	6.5444	9.616	6.5444	4.2934
4	2.8915	4.896	6.5444	4.896	2.8915
5	0	2.8915	4.2934	2.8915	0

Table 24 – The z-coordinates of a fabricated 5-by-5 surface using WaterClear 10120 resin displaced in the center by 24.576 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	10.403	13.174	10.121	0
2	10.941	16.756	19.744	17.03	10.955
3	12.927	19.94	24.579	19.928	13.183
4	10.045	16.75	20.055	17.18	11.148
5	0	10.472	13.176	10.447	0

	1	2	3	4	5
1	0	7.4376	10.915	7.4376	0
2	7.4376	12.473	16.645	12.473	7.4376
3	10.915	16.645	24.579	16.645	10.915
4	7.4376	12.473	16.645	12.473	7.4376
5	0	7.4376	10.915	7.4376	0

Hindrance of Bending Caused by the Leaf Springs Data

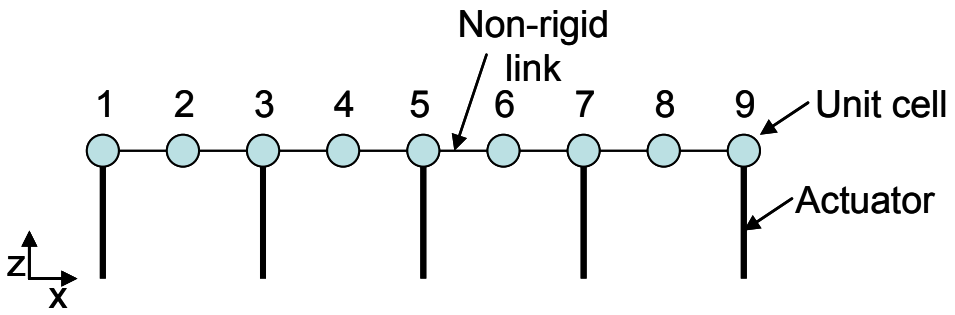


Figure 102 – A reference to the unit cell orientation depicted in the tables that follow

Table 25 – Data comparison of a 1-by-9 array of unit cells using the mathematical model (labeled MATLAB positions) and a manufactured array (labeled crust positions) where the third, fifth, and seventh unit cell were displaced

MATLAB POSITIONS			CRUST POSITIONS		
x	y	z	z1	z2	ave z
0	0	29.028	28.830	29.573	29.2015
18.95	0	29.962	29.758	30.525	30.14108
37.9	0	30	30	30	30
56.85	0	21.857	21.857	21.857	21.857
75.8	0	15	15	15	15
94.75	0	21.857	21.857	21.857	21.857
113.7	0	30	30	30	30
132.65	0	29.962	29.932	30.456	30.19424
151.6	0	29.028	28.999	29.507	29.253

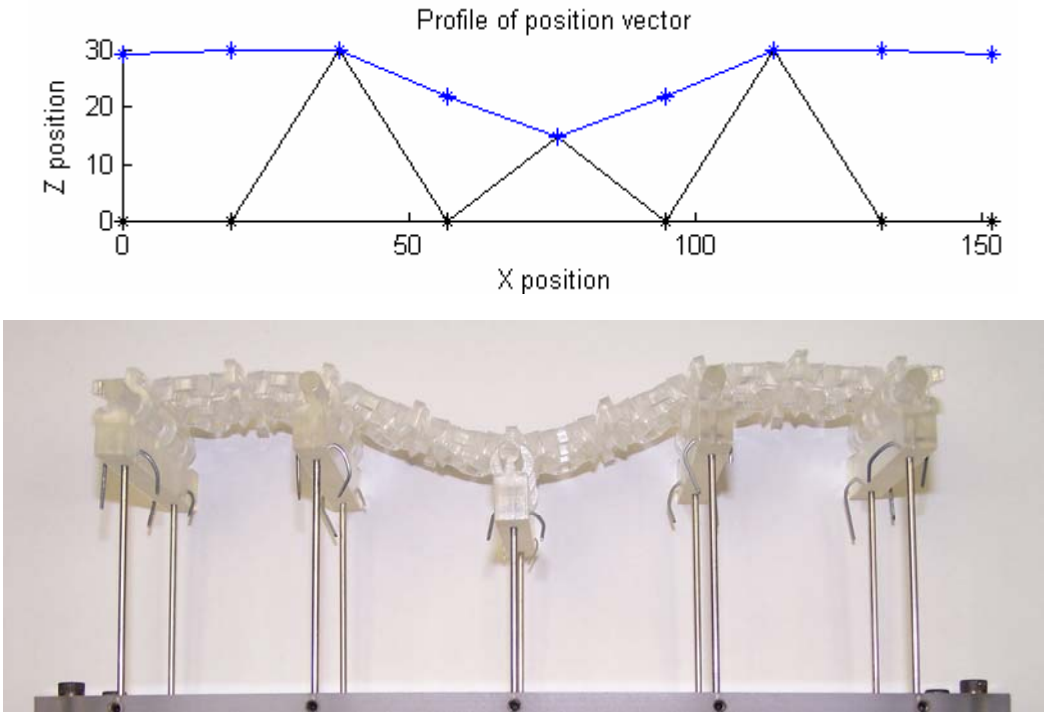


Figure 103 – An illustrative comparison of the mathematical model (top) and a manufactured 1-by-9 array (bottom) from the data set in the table above

Table 26 – Data comparison of a 1-by-9 array of unit cells using the mathematical model (labeled MATLAB positions) and a manufactured array (labeled crust positions) where the first and last unit cell were displaced

MATLAB POSITIONS			CRUST POSITIONS		
x	y	z	z1	z2	ave z
0	0	25	25	25	25
18.95	0	25.629	25.544	25.894	25.71913
37.9	0	26.16	26.073	26.431	26.252
56.85	0	26.769	26.712	27.085	26.89869
75.8	0	27.35	27.292	27.673	27.4825
94.75	0	27.999	27.967	28.144	28.05525
113.7	0	28.62	28.587	28.768	28.6775
132.65	0	29.264	29.230	29.415	29.32279
151.6	0	30	30	30	30

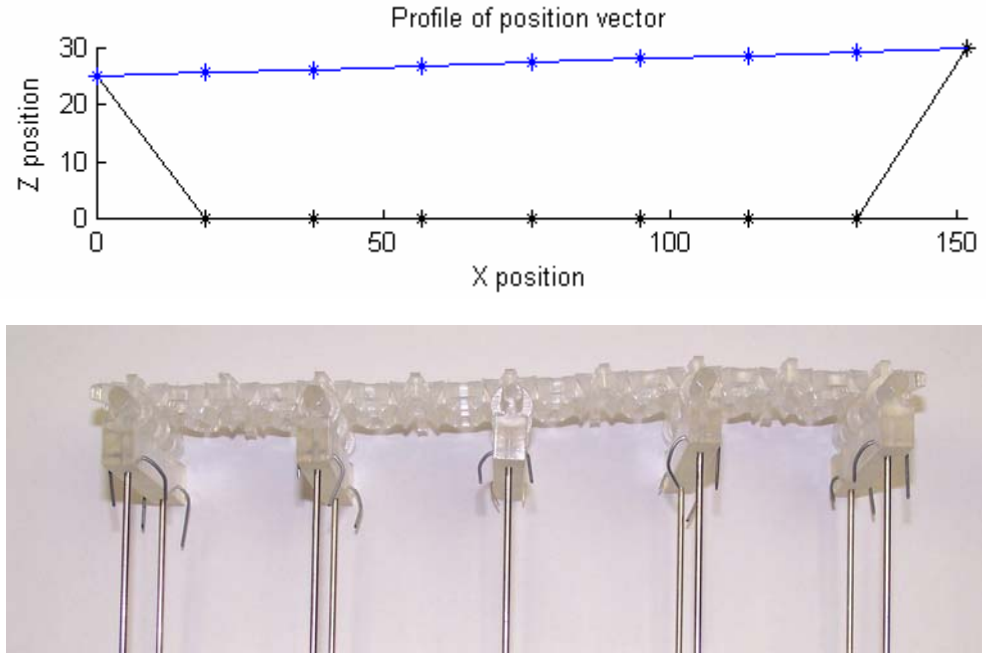


Figure 104 – An illustrative comparison of the mathematical model (top) and a manufactured 1-by-9 array (bottom) from the data set in the table above

Table 27 – Data comparison of a 1-by-9 array of unit cells using the mathematical model (labeled MATLAB positions) and a manufactured array (labeled crust positions) where the fifth, and seventh unit cell were displaced

MATLAB POSITIONS			CRUST POSITIONS		
x	y	z	z1	z2	ave z
0	0	4.6924	4.59	5.27	4.93
18.95	0	6.4039	6.264	7.192	6.728162
37.9	0	7.8212	7.256	8.321	7.7885
56.85	0	9.858	9.146	10.488	9.816784
75.8	0	10	10	10	10
94.75	0	19.965	19.965	19.965	19.965
113.7	0	30	30	30	30
132.65	0	29.983	29.180	30.610	29.89515
151.6	0	28.499	27.736	29.095	28.4155

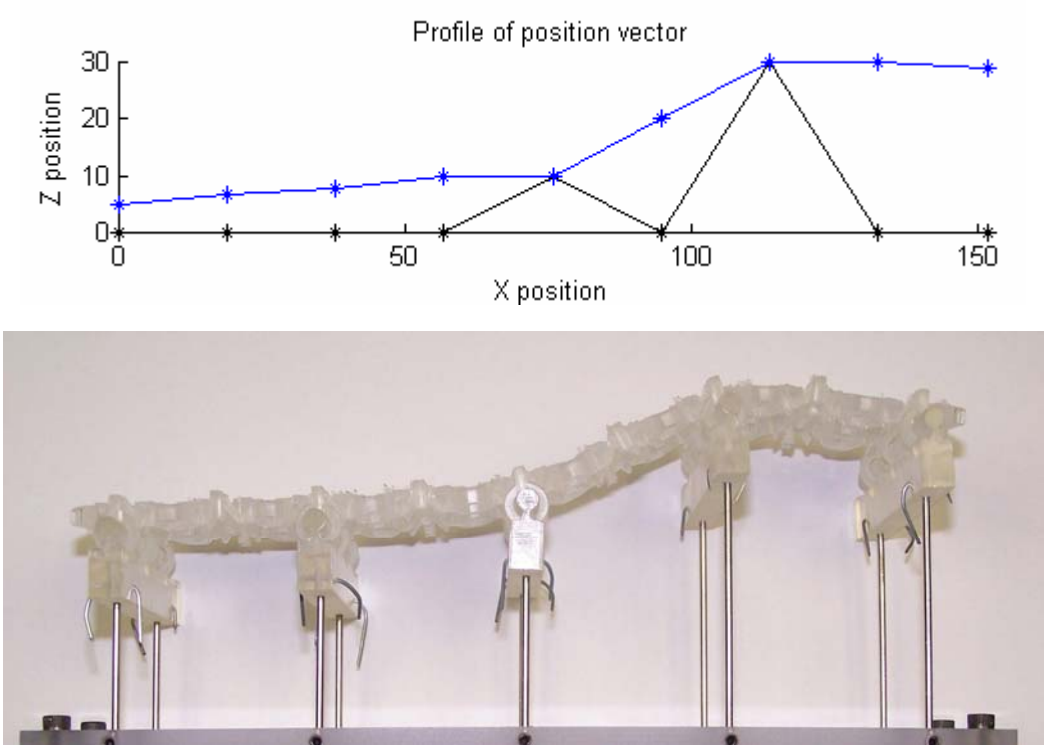


Figure 105 – An illustrative comparison of the mathematical model and a manufactured 1-by-9 array from the data set in the table above

Second Comparison Test Data Sets

Third Redesigned Unit Cell Data

Table 28 – The z-coordinates of a fabricated 5-by-5 surface with redesigned unit cells using WaterClear 10120 resin displaced in the center by 12.672 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	3.628	5.526	3.626	0
2	3.632	6.344	8.632	6.351	3.629
3	5.558	8.621	12.672	8.626	5.563
4	3.625	6.348	8.621	6.356	3.619
5	0	3.631	5.562	3.629	0

	1	2	3	4	5
1	0	3.7992	5.638	3.7992	0
2	3.7992	6.433	8.612	6.433	3.7992
3	5.638	8.612	12.672	8.612	5.638
4	3.7992	6.433	8.612	6.433	3.7992
5	0	3.7992	5.638	3.7992	0

Table 29 – The z-coordinates of a fabricated 5-by-5 surface with redesigned unit cells using WaterClear 10120 resin displaced in the center by 18.143 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	5.448	7.985	5.438	0
2	5.466	9.356	12.725	9.418	5.496
3	8.052	12.732	18.143	12.742	7.992
4	5.494	9.402	12.737	9.369	5.483
5	0	5.138	8.015	5.138	0

	1	2	3	4	5
1	0	5.454	8.063	5.454	0
2	5.454	9.212	12.308	9.212	5.454
3	8.063	12.308	18.143	12.308	8.063
4	5.454	9.212	12.308	9.212	5.454
5	0	5.454	8.063	5.454	0

Final Redesigned Unit Cell Data

Table 30 – The z-coordinates of a fabricated 5-by-5 surface with redesigned unit cells using WaterClear 10120 resin displaced in the center by 5.932 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	1.779	2.641	1.767	0
2	1.761	3.033	4.051	3.031	1.757
3	2.657	4.046	5.932	4.053	2.67
4	1.763	3.021	4.049	3.026	1.759
5	0	1.781	2.645	1.801	0

	1	2	3	4	5
1	0	1.7743	2.641	1.7743	0
2	1.7743	3.012	4.036	3.012	1.7743
3	2.641	4.036	5.932	4.036	2.641
4	1.7743	3.012	4.036	3.012	1.7743
5	0	1.7743	2.641	1.7743	0

Table 31 – The z-coordinates of a fabricated 5-by-5 surface with redesigned unit cells using WaterClear 10120 resin displaced in the center by 15.43 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	4.645	7.005	4.644	0
2	4.667	7.894	10.679	7.924	4.647
3	6.989	10.547	15.43	10.534	7.012
4	4.666	7.946	10.676	7.912	4.639
5	0	4.648	6.995	4.655	0

	1	2	3	4	5
1	0	4.671	6.919	4.671	0
2	4.671	7.887	10.504	7.887	4.671
3	6.919	10.504	15.43	10.504	6.919
4	4.671	7.887	10.504	7.887	4.671
5	0	4.671	6.919	4.671	0

Table 32 – The z-coordinates of a fabricated 5-by-5 surface with redesigned unit cells using WaterClear 10120 resin displaced in the center by 19.207 mm (left) and the z-coordinates of the corresponding surface using the mathematical model

	1	2	3	4	5
1	0	5.802	8.563	5.901	0
2	5.829	9.763	13.119	9.595	5.803
3	8.583	13.095	19.207	12.987	8.533
4	5.928	9.634	13.171	9.741	5.89
5	0	5.867	8.499	5.902	0

	1	2	3	4	5
1	0	5.799	8.558	5.799	0
2	5.799	9.771	13.044	9.771	5.799
3	8.558	13.044	19.207	13.044	8.558
4	5.799	9.771	13.044	9.771	5.799
5	0	5.799	8.558	5.799	0

REFERENCES

[1] M. Allen, W. Book, I. Ebert-Uphoff, A. Glezer, D. Rosen, and J. Rossignac. "Digital Clay for Shape Input and Display." NSF-ITR/PE+SY Award 0121663, April 2001.

[2] H. Zhu, and W. Book. "Embedding and Multiplexing Large Scale Sensor Arrays for Digital Clay." Proceedings of ASME-IMECE, Orlando, FL, Nov 5-11, 2005.

[3] P. Bosscher, and I. Ebert-Uphoff. "Digital Clay: Architecture designs for shape-generating mechanisms." IEEE International Conference on Robotics and Automation, May 2003.

[4] H. Zhu, and W. Book. "Control Concepts for Digital Clay." IFAC Symposium on Robotics and Control, September 2003.

[5] J. Fritz, T. Way, and K. Barner. "Haptic Representation of Scientific Data for Visually Impaired or Blind Persons." 11th Annual CSUN Technology and Persons with Disabilities Conference, 5 pages, Northridge, California, 1996.

[6] J. Fritz, and K. Barner. "Design of a Haptic Data Visualiztion System for People with Visual Impairments." IEEE Transaction on Rehabilitation Engineering Sept 1999: 372-384.

[7] M. Shinohara, Y. Shimizu, and A. Mochizuki. "Three-Dimensional Tactile Display for the Blind." IEEE Transaction on Rehabilitation Engineering Sept 1998: 249-256.

[8] T. James, G. Humphery, J. Gati, P. Servos, R. Menon, and M. Goodale. "Haptic study of three-dimensional objects activates extrastriate visual areas." Neuropsychologia 40 2002: 1706-1714.

[9] C. Ramstein. "Combining Haptic and Braille Technologies: Design Issues and Pilot Study. 1996. Proc. ASSETS'96, ACM, pp. 37-44, Vancouver, Canada.

[10] Bosscher, Paul, and Imme Ebert-Uphoff. "A Novel Mechanism for Implementing Multiple Collocated Spherical Joints," Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, Sept 14-19, 2003.

[11] Nguyen, “Designing, Manufacturing, and Predicting Deformation of a Formable Crust Matrix,” Masters Thesis, Mechanical Engineering Department, Georgia Institute of Technology, July 2004.

[12] Rosen, Nguyen, “Simulation Methods for Formable Crust Skins of Digital Clay Human-Computer Interface Devices,” Proceedings of DETC’05 ASME Computers and Information in Engineering Conference, paper DETC2005/CIE-84455, Long Beach, Sept 24-28, 2005.

[13] S.D. Laycock, and A.M. Day. “Recent developments and applications of haptic devices.” Computer Graphics Forum 22, pp. 117-132, 2003.

[14] P. Kammermeir, M. Buss, and G. Schmidt. “Dynamic Display of Distributed Tactile Shape Information by a Prototypical Actuator Array.” IEEE/RSJ International Conference on Intelligent Robotics and Systems, 2000.

[15] P. Wellman, W. Peine, G. Favalora, and R. Howe. “Mechanical Design and Control of a High-Bandwidth Shape Memory Alloy Tactile Display.” Proceedings of the Fifth International Symposium on Experimental Robotics, June, 1998.

[16] G. Moy, C. Wagner, and R.S. Fearing. “A compliant tactile display for teletaction.” IEEE International Conference on Robotics and Automation, San Francisco, CA, 2000.

[17] T. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill. “A Hand Gesture Interface Device.” Proc. Human Factors in Computing Systems and Graphics Interface, ACM Press, New York, April 1987, pp. 189-192.

[18] Virtual Technologies, “CyberGrasp: Groundbreaking Haptic Interface for the Entire Hand,” http://www.virtex.com/products/hw_products/cybergrasp.html (Accessed September 9, 2006).

[19] M. Benali-Khoudja, M. Hafez, J. Alexandre, and A. Kheddar. “Tactile interfaces: a state-of-the-art survey.” International Symposium on Robotics, 2004.

[20] K. Hale and K. Stanney. “Deriving haptics guidelines from human physiological, psychophysical, and neurological foundations. IEEE Computer Graphics and Applications, March, 33-39.

[21] SensAble Technologies, “Haptic Devices,” http://www.sensable.com/products-haptic-devices.htm (Accessed September 9, 2006).

[22] T. Massie, and K. Salisbury. “The PHANToM Interface: A Device for Probing Virtual Objects.” ASME Winter Annual Meeting, DSC-Vol.55-1. 1994.

[23] I. Poupyrev, M. Okabe, and S. Maruyama. “Haptic Feedback for Pen Computing: Directions and Strategies.” Extended Abstracts of the ACM Conference on Human Factors in Computing Systems, CHI 2004. ACM, New York, NY, pp. 1309-1312.

[24] H. Iwata, H. Yano, F. Nakaizumi, and R. Kawamura. “Project FEELEX: adding haptic surface to graphics.” Proceedings of SIGGRAPH2001, 2001.

[25] P.M. Taylor, A. Moser, and A. Creed. “A sixty-four element tactile display using shape memory alloy wires.” *Displays* **18**, pp. 163-168, 1998.

[26] V. Hayward, and J.M. Cruz-Hernandez. “Tactile Display Device Using Distributed Lateral Skin Stretch.” In Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. IMECE 2000 Conference, Nov. 5-10, 2000.

[27] R. Perry, and S. Frisken. “Kizamu: A system for sculpting digital characters.” Proceedings of ACM SIGGRAPH2001, pp. 47-56, 2001.

[28] E. Schweikardt, and M. Gross. “Digital Clay: Deriving digital models from freehand sketches.” Proceedings of ACADIA 1998, Quebec City, Canada, pp. 202-211, Oct. 1998.

[29] D. Rosen, “Chapter 1: Introduction to Additive Manufacturing Processes,” Lecture notes for ME 7227, Georgia Institute of Technology, Atlanta, GA, Fall Semester, 2006.

[30] P. Jacobs. *Stereolithography and other RP&M Technologies*. Dearborn: Society of Manufacturing Engineers, 1992.

[31] P. Jacobs. *Rapid Prototyping & Manufacturing: Fundamentals of Stereolithography.* Dearborn: Society of Manufacturing Engineers, 1996.

[32] J. Beaman, J. Bourell, D.L. Bourell, R.H. Crawford, H. Marcus, K. McAlea. *Solid Freeform Fabrication: A New Direction in Manufacturing.* Kluwer Academic Publishers, Boston, 1997.

[33] R. Solanki, C.A. Moore, and G.J. Collins. "Laser-Induces Chemical Vapor Deposition." *Solid State Technology,* pp. 220-226, June 1985.

[34] J. Maxwell, M. Boman, K. Williams, K. Larsson, N. Jaikumar, and G. Saiprasanna. "High-Speed Laser Chemical Vapor Desposition of Amorphous Carbon Fibers, Stacked Conductive Coils, and Folded Helical Springs." SPIE Conference on Micromachining and Microfabrication Process Technology V, SPIE vol. 3874, Santa Clara, CA, 1999.

[35] Limaye, "Design and Analysis of a Mask Projection Micro-stereolithography System." Masters Thesis, Mechanical Engineering Department, Georgia Institute of Technology, Nov 2004.

[36] Bosscher, Paul, and Imme Ebert-Uphoff. "A Novel Mechanism for Implementing Multiple Collocated Spherical Joints." Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, Sept 14-19, 2003.

[37] G. Becker. "Special Problem: Continued Development of the 'Formable Crust' Design and its Manufacturability." Final Report of ME8913, Mechanical Engineering Department, Georgia Institute of Technology, Aug 2005.

[38] Y.-M. Moon, B.T. Trease, S. Kota, "Design of Large-Displacement Compliant Joints," Proceedings of DETC'02 ASME Computers and Information in Engineering Conference, paper DETC2002/MECH-34207, Montreal, Canada, Sept 2002.

[39] S.C. Chapra, and R.P. Canale. *Numerical Methods for Engineers.* 3rd edition. New York, NY: McGraw-Hill, 1998.

[40] The MathWorks, "Optimization Toolbox: lsqnonlin," http://www.mathworks.com/access/helpdesk_r13/help/toolbox/optim/lsqnonlin.html (Accessed September 9, 2006).

[41] J.N. Reddy. *Introduction to the Finite Element Method.* 2nd edition. New York, NY: McGraw-Hill, 1992.

[42] ANSYS, Inc, http://www.ansys.com (Accessed October 29, 2006).

[43] MatWeb, http://www.matweb.com (Accessed October 29, 2006).

[44] The IADR Pan European Federation 2006, "Correlation of Filler Content and Elastic Properties of Resin-Composites," http://iadr.confex.com/ladr/pef06/techprogram/abstract_84714.htm (Accessed October 29, 2006).

[45] Protogenic, "SLA Resin Specifications," http://protogenic.com/admin/images/sla_resins.pdf (Accessed September 7, 2006).

[46] DSM, "WaterClear 10120: Optically Clear, Rigid, Durable Epoxy Resin For Stereolithography For Solid State (355 nm) Laser Systems," www.dsm.com/en_us/html/dsms/pd_product_data_sheets.htm (Accessed September 7, 2006).

[47] 3D-Systems, "SLA 5510 Resin," www.3dsystems.ru/products/slaseries/sla5000/materials/sla5000_5510.asp.htm (Accessed September 7, 2006).

[48] Huntsman, "Stereolithography Materials: RenShape SL 5510 for use on Viper si2 SLA system," http://www.renshape.com (Accessed September 7, 2006).